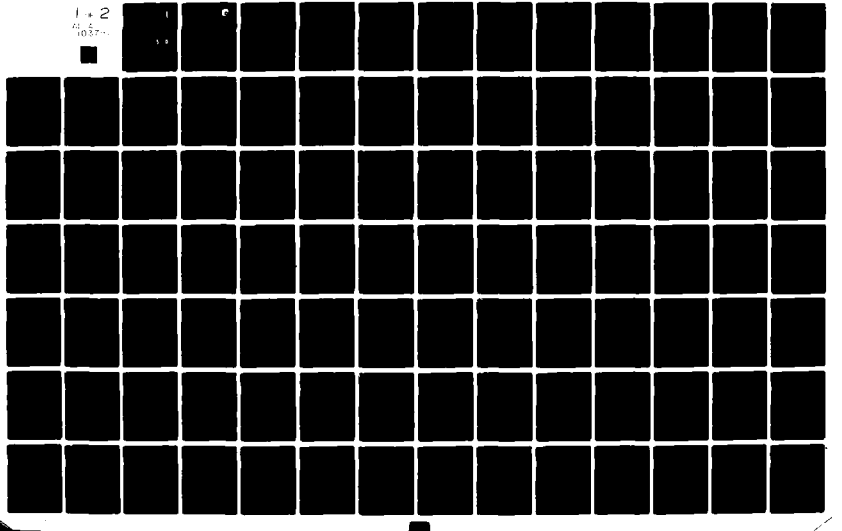
AD-A103 782 SINGER CO WAYNE NJ F/G 9/2
A USER ORIENTED MICROCOMPUTER AND MONITOR SYSTEM.(U)
FEB 81 J W STARK F33615-77-C-1176
UNCLASSIFIED AFWAL-TR-81-1130 NL



PHOTOGRAPH THIS SHEET

AD A103782

DTIC ACCESSION NUMBER

II

LEVEL

Singer Co., Wayne, NJ

I

INVENTORY

A User Oriented Microcomputer And Monitoring System. Final Rept., 15 Feb. 81

DOCUMENT IDENTIFICATION

Contract F33615-77-C-1176 AFWAL-TR-81-1130

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR	
NTIS	GRA&I
DTIC	TAB
UNANNOUNCED	
JUSTIFICATION	
BY	
DISTRIBUTION /	
AVAILABILITY CODES	
DIST	AVAIL AND/OR SPECIAL
A	

DISTRIBUTION STAMP

DTIC
ELECTE
SEP 4 1981
S D
D

DATE ACCESSIONED

PROPRIETARY Statement on page 101, fig. 30, of subject report, Deleted, Per Ltr. on file.

DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

DTIC FORM OCT 79 70A

DOCUMENT PROCESSING SHEET

AFWAL-TR-81-1130

A USER ORIENTED MICROCOMPUTER AND MONITOR SYSTEM

John W. Stark
The Singer Company
150 Totowa Road
Wayne, New Jersey 07470

TECHNICAL REPORT AFWAL-TR-81-1130

15 February 1981
Final Report for Period October 1977 - June 1981

AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Guy Vince

GUY VINCE
Electrical Engineer
AFWAL/AAAT-2

Donald L. Moon

DONALD L. MOON, Chief
Information Processing Technology Branch
Avionics Laboratory

FOR THE COMMANDER

Raymond E. Siferd

RAYMOND E. SIFERD, COL, USAF
Chief, System Avionics Division
Avionics Laboratory

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWA/AAAT-2, *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWAL-TR-81-1130	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A User Oriented Microcomputer And Monitoring System		5. TYPE OF REPORT & PERIOD COVERED Oct 77 - Jun 81 Final Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John W. Stark		8. CONTRACT OR GRANT NUMBER(s) F33615-77-C-1176
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Singer Company 150 Totowa Road Wayne, New Jersey 07470		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2003 0417
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Avionics Laboratory Air Force Systems Command Wright-Patterson AFB, Ohio 43433		12. REPORT DATE 15 February 1981 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Microcomputer, Monitor, MIL-STD-1750, MIL-STD-1553B, Program Testing, Program Performance Measurement, Input Output Simulation, Trace, Symbol Tables, Common File Formats

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This Report documents the design definition, implementation, and test of a User Oriented MIL-STD-1750 Microcomputer and Monitor System. The system design incorporates several experimental features to enhance program development and measurement in an avionics integration laboratory environment, the most noteworthy of which is a monitor real time memory map function.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

LIST OF TABLES

INTRODUCTION

On 77 October 01 Contract F33615-77-C-1176 was awarded to The Singer Company by the United States Air Force, AFSC Aeronautical Systems Division, Wright-Patterson AFB, Ohio, 45433, to conduct research and development leading to the delivery of a breadboard Microcomputer/Monitoring System to The Air Force Avionics Laboratory. On 80 June 20 a Computer and Support System was delivered to the Air Force by The Singer Company. This System, the User Oriented Microcomputer and Monitoring System, UOMMS, is discussed in this Final Report.

The goal of the program was to develop a low-cost computer and the necessary support system (User's Console with built-in software development, checkout, and performance monitoring facility) that would reduce the time and cost of developing, verifying, and maintaining an embedded computer system, hardware and software.

During the life of the contract the design evolved with concurrent Air Force standardization efforts. The final Microcomputer implements the instruction set architecture of MIL-STD-1750 and complies with MIL-STD-1553B. The Monitor is unique to the UOMMS Program.

The program scope included in sequence a requirements analysis phase, system design phase, build and test phase, and a final acceptance and evaluation phase at WPAFB.

REQUIREMENTS ANALYSES

During the initial phase of the contract various analyses, investigations, and surveys were undertaken to define requirements for the Microcomputer and the Monitor. Concurrently, parts availability for microcomputer implementation was surveyed. The established time frame for applicability of this work was 1977 through 1982.

Specific areas covered were: (1) Computational Requirements for the Microcomputer, (2) Microcomputer Instruction Set Considerations, (3) Part Sets Selection, (4) Microcomputer Internal Communications, and (5) Monitor Features.

Computational Requirements

A significant consideration in the system definition was what computational requirements might be imposed over the time frame of interest. An analysis was undertaken to roughly estimate the expected throughput and storage requirements.

The methodology chosen for this analysis was the use of functional mixes to estimate required throughput. Eventually the twelve avionics functions listed below were selected for evaluation:

1. Inertial Navigation
2. Strapdown Navigation
3. Hybrid Navigation
4. Weapon Delivery Air/Ground
5. Weapon Delivery Air/Air
6. Air Data Computer
7. Loran
8. Integrated Tactical Navigation
9. Terrain Matching
10. Display Processing
11. Symbol Generation
12. Flight Control.

Depending on availability in late 1977, existing operational programs or existing developmental programs were examined. Each instruction discovered was allocated to one of the following categories:

1. Load/Add/Store 16 Bit Fixed Point
2. 16 by 16 Multiply 16 Bit Fixed Point
3. 32 by 16 Fixed Point Division
4. Load/Store 32 Bit Fixed Point
5. 32 by 32 Fixed Point Multiply
6. Floating Point Add/Subtract Commands (32 Bit Word, 24 Bit Mantissa, 6 Bit Align/Normalization)
7. Floating Point Multiply
8. Square Root Calls 32 Bit Argument
9. Sin/Cos Calls 16 Bit Argument
10. Arctan Calls 16 Bit Argument.

These particular commands were chosen for generality across instruction sets and utility in avionics applications. The number of operations per second required was determined from these values and from known or hypothesized iteration rates.

Next, four memory mixes were postulated based on the following access times:

a. Fast Memory-Semiconductor with 75ns Access
b. Medium Memory-Semiconductor with 125ns Access
c. Core Memory 350ns Access, 800ns Cycle
d. A Mix of Fast and Medium Semiconductor Memory.

Execution times for the primitives based on these times were computed.

These execution times were used to compute the throughput for each of the four memory configurations for each of the applications.

Finally, the time utilization was determined by computing the ratio of required operations per second to the number of possible operations per second for the application, for each memory type.

$$\frac{\text{Req. Operations/Sec}}{\text{Possible Operation/Sec}} \times 100 = \text{Utilization}$$

Table 1, Computer Resource Requirements, Circa 1977 summarizes the throughput and memory (16 bit words) requirements generated by the study. Figure 1, Throughput Versus Functions, show the relationships between requirements for the applications examined for both the present and near term future.

Instruction Set Considerations

The Instruction Set of the existing DAIS Computer was examined for possible improvements in programming efficiency. Some of the recommendations are discussed in the following paragraphs.

1. Delete overflow/underflow bits from the status word replacing the function by separate internal interrupts to identify the cause. This suggestion was followed.

2. Eliminate explicit specification of status word bits for arithmetic tests. This suggestion was not followed.

3. Remove even word restriction for double words. This suggestion was followed except for base relative formats where field space was at a premium.

TABLE 1. COMPUTER RESOURCE REQUIREMENTS - CIRCA 1977

FUNCTION	CURRENT KOPS	MEMORY (16 BIT WORDS) R/W	DATA	INSTRUCTIONS
INERTIAL NAVIGATION	89.3	1024	2560	5692
STRAPDOWN NAVIGATION	108.9	1024	2560	6500
HYBRID NAVIGATION	98.5	1024	2560	5948
WEAPON DELIVERY AIR/GROUND	119.5	2048	3072	11264
WEAPON DELIVERY AIR/AIR	285.8	2560	4096	15384
AIR DATA COMPUTER	363.0	512	----	1024
LORAN	3.4	512	----	2350
ITNS	274.2	4096	2048	10240
TERRAIN MATCHING	188.5	12288	4096	6144
DISPLAY PROCESSING	151.6	8192	----	4192
SYMBOL GENERATOR	4000.0	4096	----	4192
DIGITAL FLIGHT CONTROL	196.7	1024	----	6144

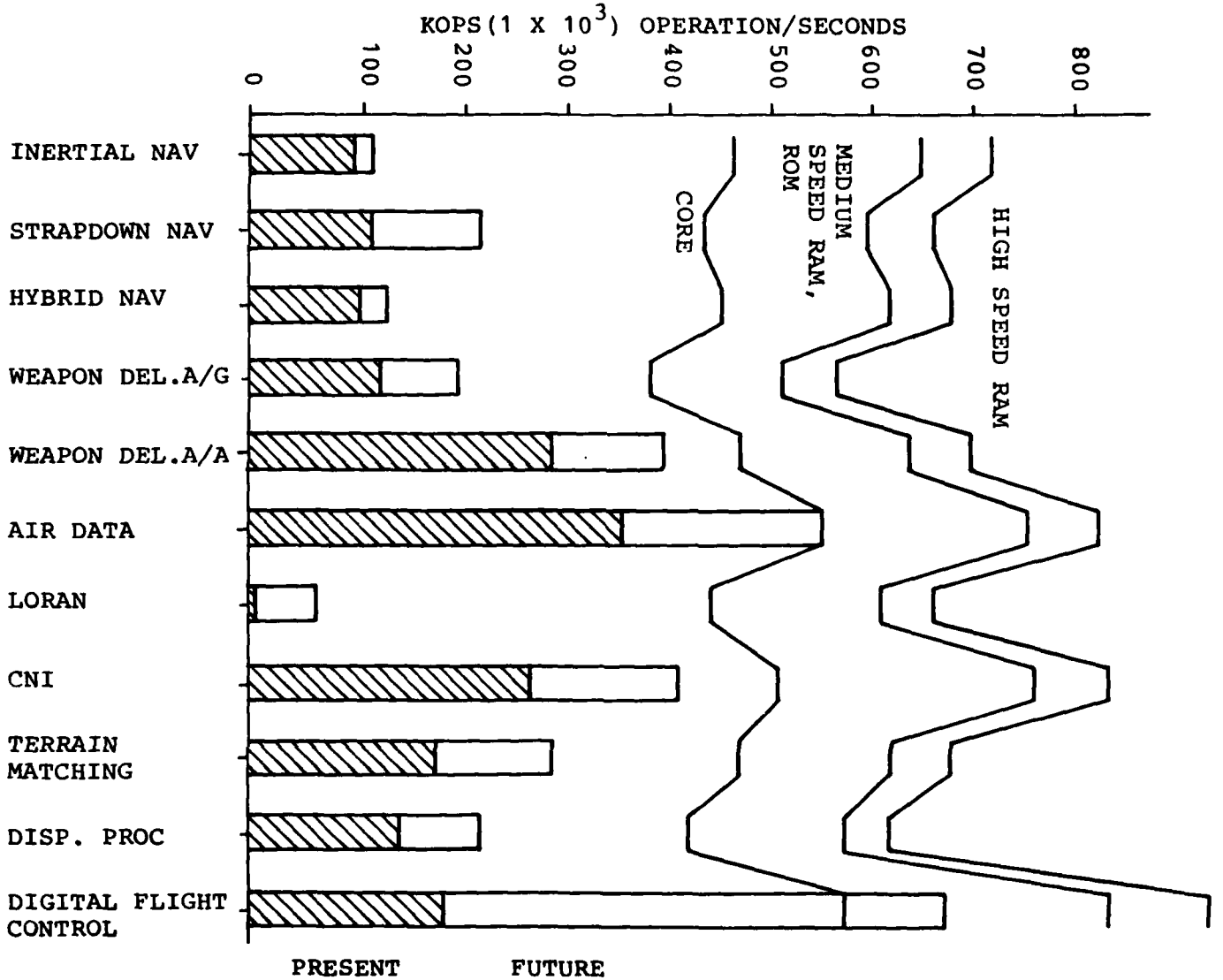


FIGURE 1. THROUGHPUT VERSUS FUNCTIONS

4. The fixed point to floating point and floating point to fixed point conversions instructions should consider using 32 bits for increased precision. Instructions were developed for 16 bit, 32 bit integer, and floating and extended floating words.

5. Re-evaluate the utility of byte operations, particularly from the compiler perspective. Byte operations were retained.

6. Replace the uninterruptible block move instruction with an interruptible version. This suggestion was followed.

7. Review the utility of the execute operation. It was deleted.

8. Delete the jump back and count instruction. An instruction conceptually similar was implemented (subtract one and jump).

9. Separate the power-on location from the interrupt area. The start-up location is no longer specified.

10. Provide careful review of procedure call mechanism. No changes implemented.

11. Make provision for position independent code via long relative jumps and/or indirect jumps. Indirect jumps were incorporated.

12. Consider the addition of a full word immediate capability. This was incorporated.

13. Simplify conversion from extended precision floating point format to floating point format by making the first two words of the expended precision floating point representation the same as the floating point format.

14. Consider simplifying the interrupt logic for the microcomputer. This was not done.

15. Consider the inclusion of:

    a. Inverse Floating Subtract and Divide
    b. Floating Square Root
    c. Bit Toggle.

    A bit set, bit reset, and bit test and set function are now included.

Parts Set Selection

The effort was directed toward finding a basic family of available parts capable of implementing a bit sliced microprogrammable processor. While the instruction set architecture was not fully defined at the time of the study the following baseline characteristics were assumed:

a. 16 General Registers

b. 4 Primary Addressing Modes

   Register to Register
   Memory Direct or Indexed
   Base Relative
   Immediate

c. Addressing Range 65536 Words (16 Bits)

d. Arithmetic

Fixed Point	Single Precision (16 Bits) Double Precision (16 Bits)
Floating Point	Single Precision (32 Bits) Extended Precision (48 Bits)

The three chip sets selected for preliminary evaluation were the:

a. AMD 2900
b. Fairchild 9400, and the
c. Motorola 10800.

Preliminary designs were completed to arrive at relative parts counts, power, speed, and cost. The AMD 2900 design with 162 parts, 38.4 watts, 1.98 microsecond average execution time, and 3.1K cost was the normalizing factor. The mix employed was an 80/20 fixed point.

CPO	Parts	Power	1/Execution Time	Cost
AMD 2900	1.	1	1	1
Fairchild 9400	1.02	.98	.67	.97
Motorola 10800	1.6	2.7	1.07	1.39

The conclusion of the study was to pursue an AMD part set implementation in more detail.

Microcomputer Internal Communications

As part of the requirements review, various existing bus architectures were reviewed in order to determine the communications mechanism for the microcomputer.

Items that were considered in the study were:

a. Bus Control Protocols
b. Transfer Control
c. Monitor Interfacing, and
d. Bit Features.

Bus control protocols refer to schemes for requesting, granting, and releasing mastership of a bus where more than one master is possible. Two basic schemes were examined, a single level distributed control mechanization and a single level centralized control mechanization. In the distributed control protocol all users shared a common request line and pass a grant signal through potential requestors. Priority is based on relative position in the grant chain. Granting is initiated by a bus release on a common release line. In the central protocol examined individual request lines are serviced by an encoded bussed grant. Each potential user must listen for his code. The distributed control requires some arbiter logic at each potential master but fewer connections in multiple master systems. Priority is based upon physical position in the grant string and hence, relatively inflexible. Attempts to balance access to the bus among masters requires additional logic at each master. For a given technology distributed control transfer is slower than centralized control since the grant is serially passed and must be retimed for each control transfer. In the centralized control flexibility is achieved at the expense of more interconnections since each potential master has an individual request line. The grant may be individual or encoded depending on the parts trade-off, which depends on the number of possible masters. For systems with a small number of masters neither approach has a significant advantage. The centralized control was chosen for implementation.

Information transfers require the transmission of an address or an address interpreted as a command from a master to memory or an input/output device followed by the transmission of a data word to or from the master. The microcomputer was intended to operate with memories and IO devices of varying response time. Rather than tieing performance to the slowest potential device with "one way control" instead, request acknowledge controls were examined. Consideration was given as to whether (1) one bus, time multiplexing address and data, (2) separate address and data busses, or (3) multiple address and data busses should be employed. The controls for address and data are fully interlocked with the appropriate acknowledge signals. Initially a

time multiplexed data/address bus was selected primarily due to the lesser number of interconnects and, in spite of, the tighter timing requirements to maintain adequate throughput. In the detail design phase this decision was modified in favor of separate data and address busses. The separate busses traded the buffering requirements of the slaves for additional drivers and receivers while allowing potentially higher speed operation on the bus with less risk. While the bus has the capability of multiword transfers this option to hold the bus is only exercised in the test and set instruction.

Some of the intended features of the overall system require real time recording of microcomputer data or near real time reaction to microcomputer input/output processing. This has been implemented by extending the buffered full microcomputer bus to the monitor.

The bus arbitration logic contains logic to monitor bus transaction time, unblock the bus, and issue a fault flag.

Monitor Features

During the study phase an investigation was performed to determine the monitor system features deemed necessary for avionics software checkout. Key objectives were:

a. To determine the characteristics of a hardware/software tool of general utility to a system's analyst,

b. To define a system that would be flexible and extendible, and

c. Determine the characteristics of a user interface that would be easy to learn and use.

The investigation consisted of analyses, feature questionnaire development, circulation, and review, and Air Force informal evaluation.

Currently, in the area of hardware/software test support, various software tools/hardware facilities have been developed. Software test facilities consisting of medium to large scale support computer and software tools (system simulators, data reduction/analysis programs, etc.) have been employed to initially test avionics software. Software test stands, consisting of limited hardware components, have been used for initial real time testing. In these configurations, the host computer would operate in real time using specialized avionics computer probe devices and avionics subsystem simulation models,

which interface with the flight computer through I/O hardware converters, to provide initial real time visibility into the avionics system. "Hot-bench" real time equipment, including a near suite of sensors, have been used to monitor system operation/interaction in the "near-live" environment. Finally, flight support facilities have been employed to monitor the avionics system in a live environment.

These techniques and facilities, however, require generalization to reduce the time and cost associated with system development and test. A higher level, more sophisticated test tool is envisioned, whereby system analysts can operate on a more general level during all phases of the system evolutionary cycle. This test tool in combination with the support software packages (compiler, assembler, linker, reformattor, etc.) form an integrated higher order development and test facility that supports hardware/software system development, test, maintenance and field support activities.

The envisioned system is a user oriented monitoring system, which enhances the usability and testability of avionics hardware/software systems. The system is a modular and configurable device that supports a variety of operating modes including: (1) real time data acquisition, (2) performance measurement, (3) I/O device simulation, (4) interactive debugging and (5) external file operations. The monitoring system provides a user oriented system interface by which the user configures the system and the tasks that are to be performed.

Data Acquisition - The system analyst, in software test mode of operation, needs a facility for capturing microcomputer state information; i.e., the contents of CPU registers and memory. This information is needed for verification of the proper operation of the system, diagnosis of a malfunction, etc.

The envisioned monitoring system provides an efficient data acquisition facility for real time microcomputer event recognition and data capture of microcomputer state information. The specified monitoring system data recording features deemed necessary for software checkout are outlined below:

> Instruction tracing in a specified memory segment; i.e., the capture of the microcomputer state variables - program counter (PC), instruction register (IR), general registers (RØ...R15), status register (SR), interrupt mask (IMR), operand address (OA), and operand (O) and selected memory locations associated with an instruction execution.

Transfer of control tracing (anywhere) or optionally, transfer of control tracing in a specified memory segment; i.e., the capture of source and destination addresses, IR, RØ...R15, IMR, SR, and selected memory locations associated with an executed jump.

Operand access tracing in a specified memory segment; i.e., operand read, write, read/write, and optionally that the operand is equal, not equal in range or out-of-range of preset fixed-point (signed integer) or floating point quantities; i.e., the capture of the PC, IR, RØ...R15, OA, O, IMR, SR, and selected memory locations associated with an operand access.

Specified instruction tracing in a specified memory segment; i.e., the capture of the PC, IR, RØ...R15, OA, O, IMR, SR, and selected memory locations associated with a specified instruction execution.

External signal tracing; i.e., the capture of the PC, IR, RØ...R15, IMR, SR, and selected memory locations based upon a signal from an external source.

Performance Measurement - The system analyst, in performance measurement mode of operation, needs information regarding the performance of the avionics system; i.e., hardware and software. This performance measurement data can be analyzed to determine the general behavior of the avionics system. In addition, he needs detailed information from which he can gain insight into the actual performance of mission software to facilitate software fine tuning operations.

The specified monitoring system performance measurement features deemed necessary for software checkout are outline below.

Measurement of system performance to determine:

* Major cycle time loading
* KOPS (and IOKOPS) rate
* Instruction mixes
* Memory access rate (i.e., programmed read/write)
* Memory bus utilization
* Interruption rate
* DMA transfer rate

Measurement of mission software performance to determine:

* Time loading vs. cycle
* Time loading vs. memory
* Memory access rate vs. memory
* Instruction frequency vs. memory
* Time loading vs. instruction

I/O Simulation - The system analyst, in software testing mode of operation, needs a facility for controlled, closed loop testing of mission software.

The monitoring system should provide a flexible I/O simulation facility for convenient Programmed I/O and DMA model connection, invocation and disconnection. The specified monitoring system I/O simulation features deemed necessary for software checkout are outlined below:

Enable/disable programmed I/O

Connection pf PIO modeling routines in the PDP-11 and invocation of these routines to complete PIO transactions

Connection of a DMA modeling routine in the PDP-11 and invocation of the routine based on an interval timer interrupt from the front-end

Disconnection of PIO and DMA modules

The performance characteristics of this conceptual tool are outlined below.

1. The system shall provide support for real time DMA transfers controlled by the front-end to (from) the microcomputer.

2. The system shall provide support for non-real time PIO simulations. In this regard, the microcomputer external real world shall be operative or inoperative as selected by the user. Also, a unique device modeling capability shall be provided.

Interactive Debugging - The system analyst, when operating in standalone, uniprocessor of multiprocessor environments, needs a control/service facility for interactive debugging of mission software.

The envisioned monitoring system provides an interactive debugging facility for convenient microcomputer startup/termination and microcomputer state definition/display operations. The specified monitoring system features deemed necessary for software checkout are outlined below:

Unconditional startup of the microcomputers in the network

Unconditional termination of the microcomputers in the network or conditional termination based upon any one of the following conditions:

Any event that triggers data recording
A specified address appearing on the memory address bus
A specified instruction count
A specified elapsed execution time

Microcomputer singlestep operation

Microcomputer program load/unload operations

Microcomputer assembly/deassembly operations

Microcomputer register examination/modification

Other desirable characteristics of this conceptual tool are:

1. The system shall provide basic support for symbolic address evaluation; i.e., evaluations to determine the base address of procedures, structures (e.g. block tables) or unstructured items.

2. The system shall employ basic attribute information from the dictionary (i.e. as is readily available) to define a memory map and direct the formatting of deassembled information.

3. The system shall support references to a based (i.e. pointed to structure or non-structure) object. In this case, the user shall explicitly specify the base value. Based data includes the following: based variables, formal parameters and variables defined by external reference.

4. The system shall provide an extensibility feature to facilitate convenient enhancements to the monitoring system's J73/I-assembler symbol dictionary access mechanism. In this regard, future enhancements to the monitoring system for the purpose of using comprehensive attribute information and supporting references to objects within a structure shall be possible.

File Operations - The system analyst, when operating in standalone, uniprocessor or multiprocessor environments, needs a facility for processing files.

The monitoring system should provide a flexible file definition and access facility for convenient data logging and displaying of the log file, external file transfer operations and connecting user defined data reduction programs. The specified monitoring system file support features deemed necessary for software checkout are outlined below.

In the standalone configuration, the monitoring system should support the following file operations:

Establishment and maintenance of the data recording log file

Printing or displaying the log file

Attaching a user defined processor for special processing of the log file

Copying, renaming and purging of files.

In the uniprocessor and multiprocessor configurations, the monitoring system should support the following file operations:

Establishment and maintenance of the data recording log file

Transmitting files (e.g. I/O simulation routines) to a designated front-end

Requesting files (e.g. log file) from a designated front-end

Printing or displaying the log file at the host

Attaching a user defined processor for special processing of the log file at the host

Copying, renaming, purging of files at the front-end or host.

Memory Map Operations - The system analyst, in all modes of operation - realtime software testing, performance measurement, and interactive debug, needs a facility for identifying, to the monitoring system, the specific microcomputer memory region that requires analysis.

The envisioned monitoring system provides a flexible memory map facility and support operations for convenient symbolic definition (e.g., compiler and assembler symbols) of a single instruction or data region.

As a result of the study effort, several desirable monitoring system memory map support operations have been identified. These extended monitoring system features are outlined below:

Definition of the map via a user interface in which the user symbolically names the multiple memory regions and/or points of interest,

Redefinition of the map to unmark those regions of interest,

Display or printing of the map to present its current configuration,

Clearing of the memory map; i.e., complete reset,

Inversion of the memory map to support convenient changes in operating modes,

Definition of the map through an external file of user definitions to permit convenient and rapid set up of the system,

Saving of the map on an external file for subsequent use.

Monitor System User Interface - The system analyst, in all modes of operation, needs a system interface facility for configuring and controlling the monitoring system.

The monitoring system should provide a flexible user interface that enhances the usability of the monitoring system.

This interface is based on high speed CRT menu displays, which prompt the user for the required information. The implementation shall refine and adjust the "implied" decision tree, the menu content, methods of choice selection, etc., to provide a man-machine interface that is highly interactive, easy to learn and use, and that is usable by a widely divergent group of users - from very inexperienced system users to highly experienced system analysts.

THE SYSTEM DESIGN

Initially high performance and lower performance monitors and microcomputers were investigated for implementation. The Air Force elected to have the high performance monitor and lower performance microcomputer implemented.

The design concept for the monitor was to implement around an existing Minicomputer Family taking advantage of the commercially available hardware and software. Performance would be related to the number of functions implemented and the degree of real time capability achieved.

The microcomputer implementation was to utilize existing bit slice technology, and specific parts within that technology that would be multisourced by semiconductor suppliers. Total parts count and performance was constrained by minimizing hardware intensive performance features, such as multiported memory interfaces, look-ahead queues, or specialized execution units.

The Monitor (See Figure 2)

The minicomputer eventually chosen for implementation was the DEC 11/04. The DEC 11/04 was the low end model of the PDP11 series. This allowed for use of the DEC peripherals and support software and preserved an upward compatibility should the need arise. The program unique hardware was to be an IO interface attached internally to the DEC unibus and externally to the microcomputer bus, later named the IBUS (See Figure 3).

The minicomputer configuration chosen includes a line clock, an RL01 disk subsystem with a 5.2 megaword capacity, a DMC-11 channel for linking to another minicomputer, a VT52 terminal, an LA180 printer, and the monitor I/O channel.

The monitor IO channel is commanded and interrogated through memory mapped IO commands from the minicomputer. A program interrupt is available to notify the minicomputer of completion of commands and filling of data acquisition buffers. A direct memory access path is provided into minicomputer memory for buffering acquired microcomputer data.

In addition to recording the microcomputer bus traffic the monitor has bus membership with the associated capability to control the bus and read or write microcomputer memory.

Since the IBUS and its control signals are required for real time data acquisition, a requirement for the monitor data acquisition, this same path is utilized for non-real time requirements.

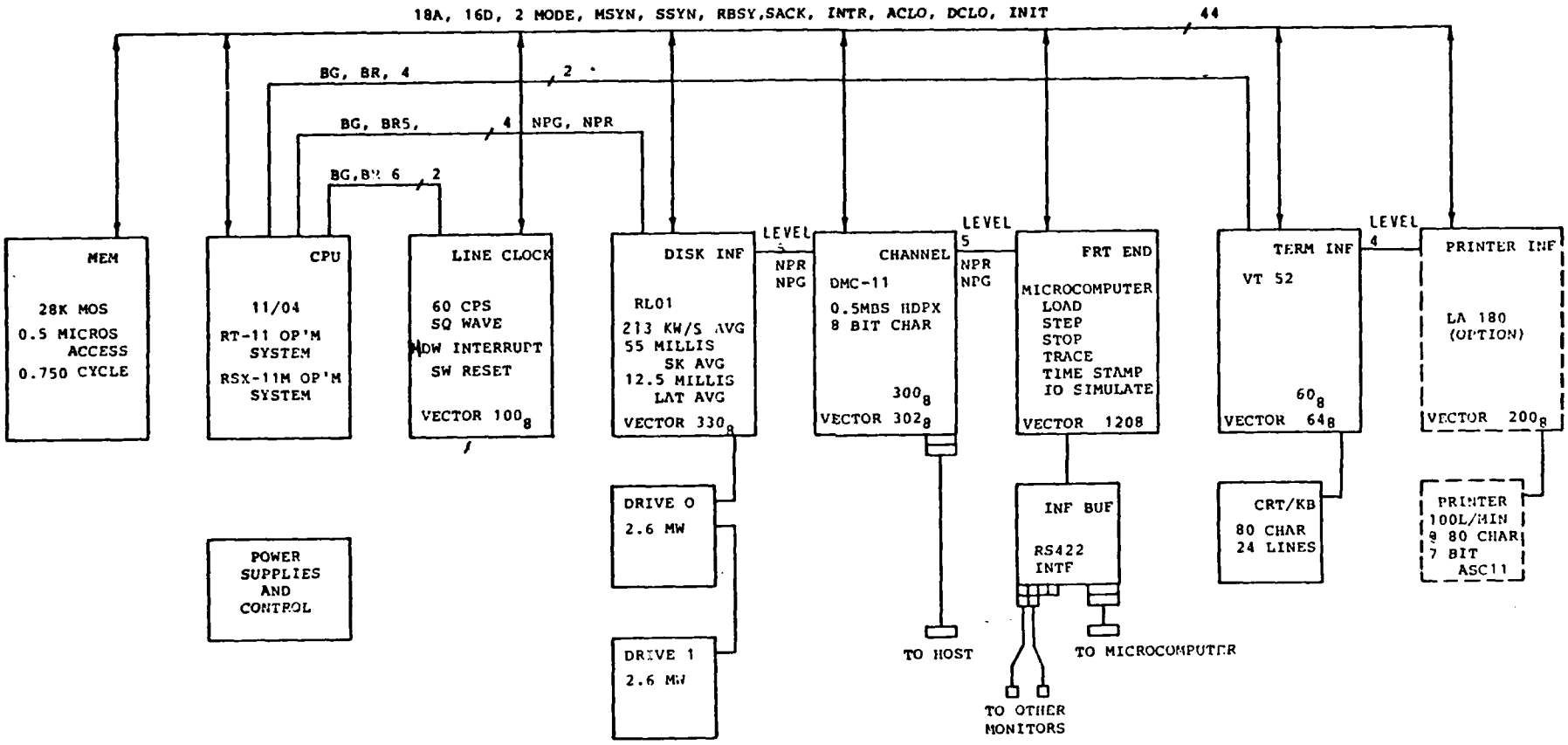


FIGURE 2. MONITOR FUNCTIONAL

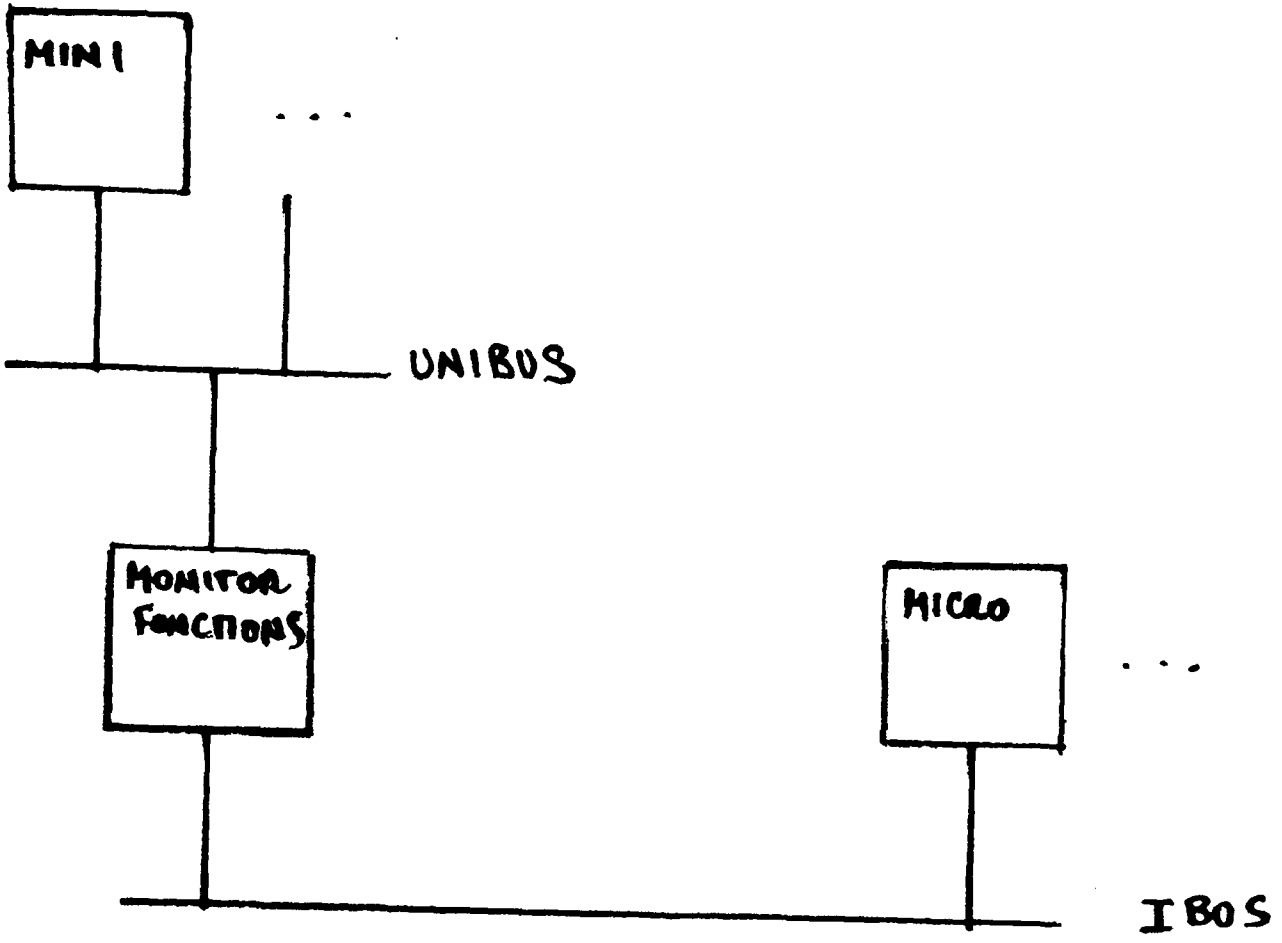


FIGURE 3. SUBSYSTEM INTERCONNECTION

Through the user interface software via prompting menu's commands and user responses internal commands are generated as memory mapped input output commands and data transfers with the monitor IO channel.

The prompting menu approach was taken to minimize the learning time of the system user by removing the requirement to learn one more control language.

After specifying a data acquisition mode, a performance measurement mode, or an IO simulation mode, the microcomputer is released to execute its program. Interactive debugging is carried out by stepping the microcomputer through one or more instructions and then examining and displaying data of interest.

The Microcomputer (See Figure 4)

The Microcomputer is composed of a bit sliced microprogrammed implementation of a MIL-STD-1750 instruction set architecture, 32K words of core memory, 8K of RAM memory, a bootload EPROM, a MIL-STD-1553B controller, an externally controlled parallel DMA channel, two interval timers, a program controller parallel input/output channel, discrete inputs and outputs, and the AGE port.

The microcomputer is capable of 1.25 million bus transactions per second. Depending on the instruction mix and memory type up to 500 or 600 thousand instructions per second may be executed.

Monitor Detail (See Figure 5)

The Monitor is implemented as a number of facilities commanded from the DEC PDP11 to load/unload, start/stop, memory simulate, input/output simulate, event and time count, and record/download microcomputer activity.

The Unibus interface in the Unibus slave mode receives commands in the form of input output mapped addresses and data to initiate interaction with the microcomputer.

For loading and reading of microcomputer memory an address and data register and Ibus control logic are provided. The monitor has the lowest priority after the microcomputer central processing unit.

The monitor possesses instruction and operand registers and sequencing logic to simulate memory in providing the instruction content and by user selection either transmit or receive operand values.

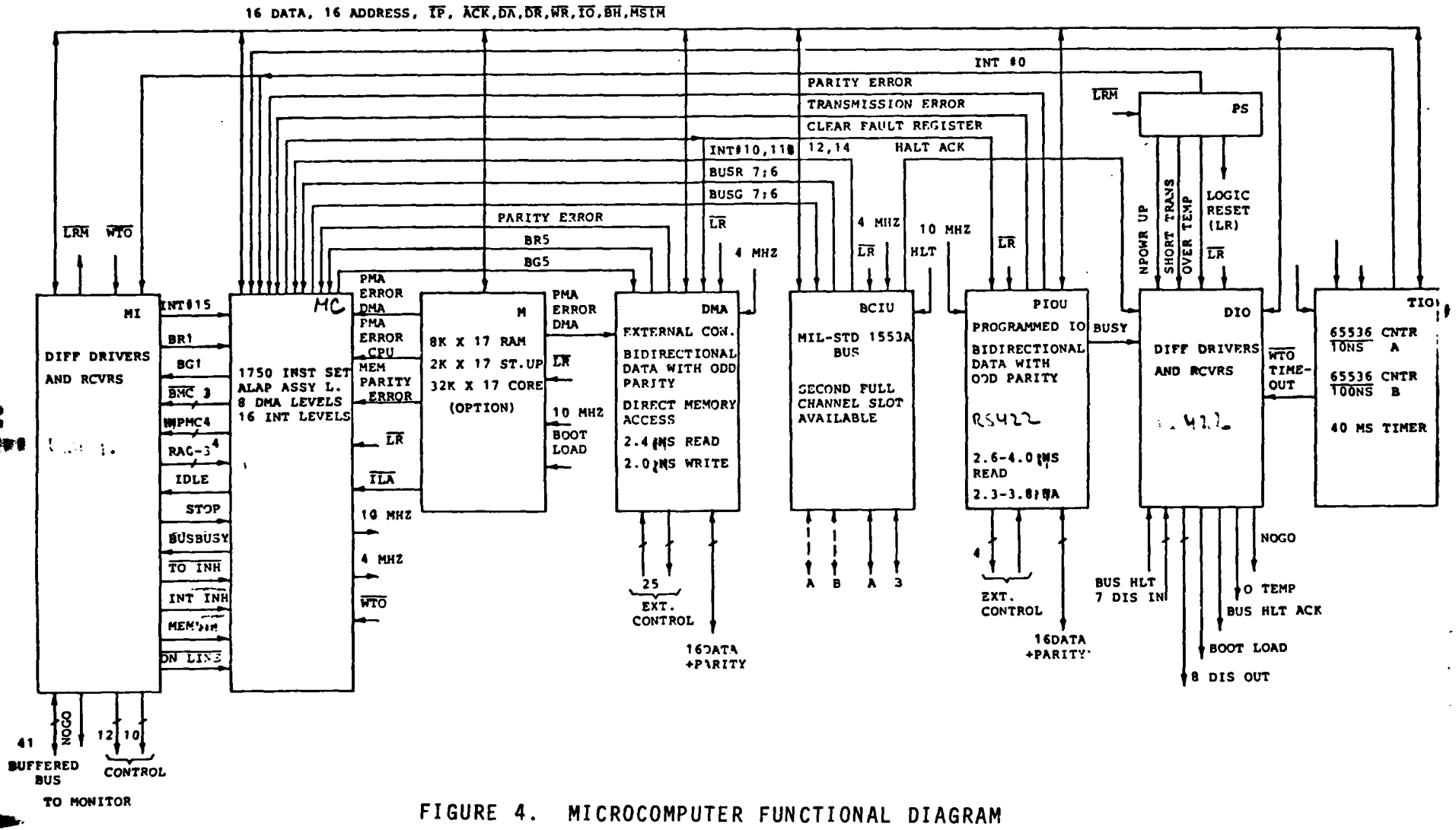


FIGURE 4. MICROCOMPUTER FUNCTIONAL DIAGRAM

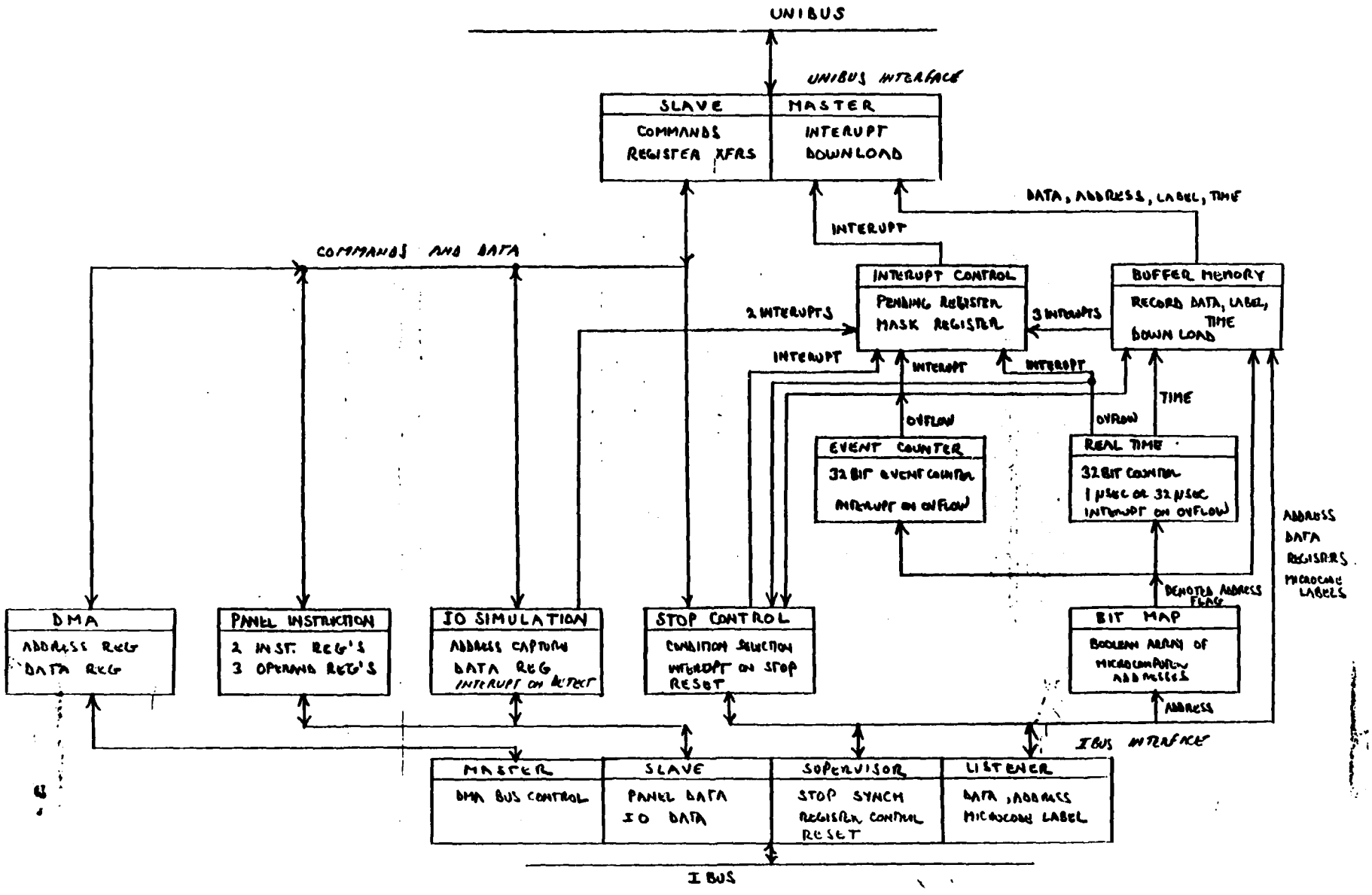


FIGURE 5. MONITOR DETAIL

Programmed input/output is simulated by capturing the microcomputer input/output address, in a Capture Register, logically holding the microcomputer bus, interrupting the DEC PDP11, and either reading or providing the requested data through a monitor register.

Stopping the microcomputer is accomplished by providing a stop signal that is sampled by the microcomputer. Upon stopping the microcomputer acknowledges with a halt signal. The halt signal in turn provides an interrupt request to the PDP11. Stopping may be initiated by direct command, recognition of a recording data rate exceeding a download rate, a logic reset, recognition of a microcomputer breakpoint or preselected address or event/time counter overflow, recognition of a monitor single step mode, or recognition of a trace event stop.

The Monitor provides a facility, the bit map, to select any number of microcomputer addresses as "denoted". The fact that a denoted address has been accessed may then be used as a condition for counting events or time or both.

Extensive logic control and buffer memory is provided as part of a microcomputer trace facility. Two buffer memories may be active to allow concurrent recording from the microcomputer and downloading to the PDP11. In addition to the microcomputer input, a label identifying the class of the recorded data, and (optionally) the time of recording is provided. Recording may be based on the occurrence of microcomputer instruction related conditions, operand related conditions, microcomputer program interrupts, DMA activity, running time or a preset number of microcomputer actions.

The downloading logic transfers blocks of sixteen words from the buffer memories to a designated area in PDP11/04 main memory. Each monitor buffer memory with data, label, and time storage uses 1536 words of RAM from which the sixteen word blocks are moved. Flags and interrupts are provided to allow downloading to start and signify its conclusion.

The monitor interrupt control contains a pending interrupt register and a mask register for fielding the various interrupt sources and providing the single interrupt to the PDP11/04.

The Ibus interface contains the drivers and receivers and the control logic to allow the monitor (1) to become a master on the Ibus, (2) to become a slave substituting for microcomputer memory or input output, (3) to force the microcomputer to reset, stop, and transmit registers, and (4) to receive data, address, and microcode label inputs.

Data communication is provided by implementing internal unidirectional busses appropriately coupled to the PDP11 Unibus and the microcomputer Ibus (see Figure 6). Unibus input data is transmitted on the Monitor D bus. Monitor output data to the Unibus is transmitted on the BFD bus. Unibus input addresses are buffered and decoded. Output addresses to the Unibus are transmitted on the ARH bus. Ibus input data is transmitted to all users except the buffer memories on the UCD bus. Ibus addresses are transmitted internally on the CA bus except for input to the buffer memories. The buffer memory data and address from the Ibus are multiplexed and transmitted on the DI bus. Output data to the Ibus is transmitted on an internal bus named Ibus.

Three internal busses are implemented to allow data flow either to the BFD bus or to the Ibus. The R bus is used with panel instructions. The LSD bus is used with input output simulation, and an unnamed bus is used with DMA.

Three other busses are the CMB bus, the M bus, and the MM bus. The CMB bus takes the Ibus data input after "anding" with masks and transmits this derived value to a number of comparators for determination of limit conditions. The M bus transmits recorded data from buffer memory #1 or buffer memory #2 to the BFD bus for downloading. The MM bus transmits label and time from buffer memory #1 or buffer memory #2 to the BFD bus for downloading.

Basic timing with the Unibus is achieved with the SSYN and NSYN signals. IP and DA are used from the Ibus. Internal to the monitor a twenty megahertz oscillator is used to generate 20MHz, 10MHz, 5MHz, 1MHz, and 31.25KHz clocks.

Unibus Interface - The Monitor operates on the Unibus as a slave to receive commands and transmit monitor status and data other than Trace data. The Monitor performs as a master on the Unibus to interrupt the PDP11/04 on bus request level five, and as a master on the non-processor request level to transmit Trace data to the PDP11/04 (see Table 2).

Slave Mode Implementation - The Monitor continuously tracks the Unibus address, control, and master synchronization for recognition of addresses between 771000 and 771176 (Octal). Upon address recognition and assertion of master synchronization the Monitor decodes the address to one of sixty-four control pulses and participates in the appropriate data transfer. See Table 3, Control Decode. Slave timing is controlled by a ten megahertz up counter released by the assertion of MSYNC in the presence of a valid address and cleared by the negation of MSYN.

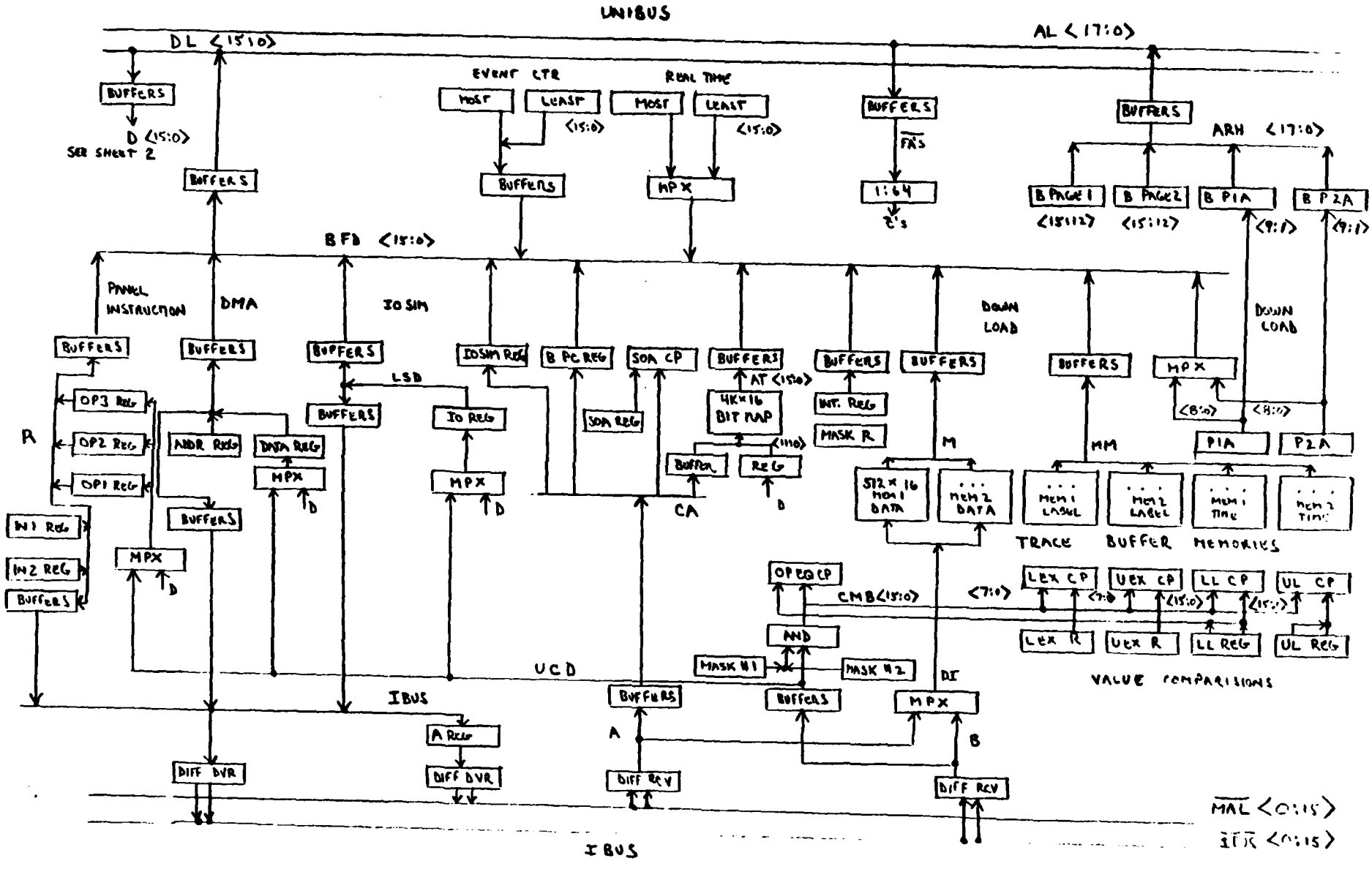


FIGURE 6. MONITOR FUNCTIONAL FLOW

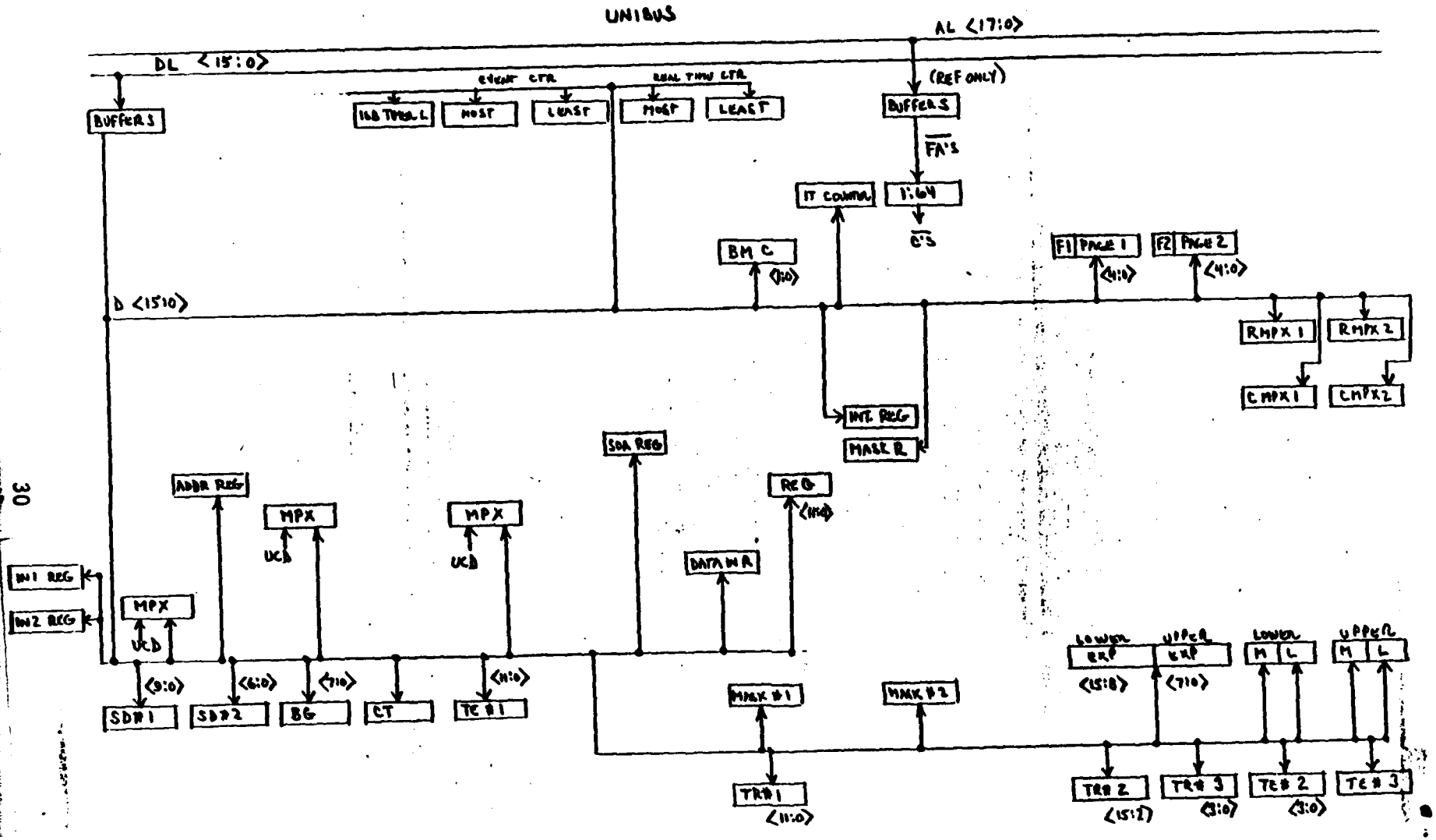


FIGURE 6. (CONTINUED) MONITOR FUNCTIONAL FLOW

TABLE 2. UNIBUS SIGNALS

DESCRIPTION	SYMBOL
Address Lines	AL 17:0
Data Lines	DL 15:0
Control	C0, C1
Master Synchronization	NSYN
Slave Synchronization	SSYN
Interrupt Synchronization	INTR
Bus Request Five	BR5
Non-Processor Request	NPR
Selection Acknowledge	SACK
Bus Busy	BBSY
Bus Grant Five	BG5
Non-Processor Grant	NPG
Initialize	INIT

TABLE 3. CONTROL DECODE

DECODE (DECIMAL)	ADDRESS (OCTAL)	FUNCTION
C1	771000	UNCONDITIONAL MICROCOMPUTER STOP REQUEST
C2	771002	LOAD MINOR STOP CONDITION SELECTION (SD#1)
C3	771004	LOAD MAJOR STOP CONDITION SELECTION (SD#2)
C4	771006	LOAD 32 BIT EVENT COUNTER LEAST SIG. HALF
C5	771010	LOAD 32 BIT EVENT COUNTER MOST SIG. HALF
C6	771012	LOAD STOP ON ADDRESS REGISTER
C7	771014	LOAD COUNTER TIMER SELECTION (CT)
C8	771016	LOAD 16 BIT TIMER PRESET REGISTER
C9	771020	LOAD 32 BIT REAL TIME COUNTER LEAST SIG. HALF
C10	771022	LOAD 32 BIT REAL TIME COUNTER MOST SIG. HALF
C11	771024	LOAD TRACE ITERATION COUNTER
C12	771026	RELEASE MICROCOMPUTER STOP
C13	771030	SPARE
C14	771032	LOAD TRACE CONDITION SELECTION (TR#1)
C15	771034	LOAD TRACE VALUE CONDITION SELECTION (TR#2)
C16	771036	LOAD TRACE DOWNLOAD MODE (TR#3)
C17	771040	LOAD UPPER LIMIT MOST SIG. WORD
C18	771042	LOAD LOWER LIMIT MOST SIG. WORD
C19	771044	LOAD UPPER LIMIT LEAST SIG. WORD
C20	771046	LOAD LOWER LIMIT LEAST SIG. WORD
C21	771050	LOAD LOWER, UPPER EXPONENT LIMITS

TABLE 3. (CONTINUED) CONTROL DECODE

DECODE (DECIMAL)	ADDRESS (OCTAL)	FUNCTION
C22	771052	LOAD INPUT DATA MASK MOST SIG. WORD
C23	771054	LOAD INPUT DATA MASK LEAST SIG. WORD
C24	771056	READ MICROCOMPUTER DMA ADDRESS
C25	771060	READ MICROCOMPUTER DMA DATA
C26	771062	RESET MICROCOMPUTER
C27	771064	LOAD DMA, PANEL INSTRUCTION SEL. (BG)
C28	771066	LOAD PANEL INSTRUCTION COMMAND WORD
C29	771070	LOAD PANEL INSTRUCTION WORD #2
C30	771072	LOAD PANEL OPERAND MOST SIG. WORD
C31	771074	LOAD PANEL OPERAND NEXT SIG. WORD
C32	771076	LOAD PANEL OPERAND LEAST SIG. WORD
C33	771100	LOAD BIT MAP DATA REGISTER
C34	771102	LOAD BIT MAP ADDRESS REGISTER
C35	771104	LOAD BIT MAP CONTROL DISCRETES
C36	771106	READ 32 BIT EVENT COUNTER LEAST SIG. WORD
C37	771110	READ 32 BIT EVENT COUNTER MOST SIG. WORD
C38	771112	LOAD MONITOR MEMORY #1 PAGE REGISTER & FLAG
C39	771114	LOAD MONITOR MEMORY #2 PAGE REGISTER & FLAG
C40	771116	RESET MONITOR AND MICROCOMPUTER
C41	771120	READ 32 BIT REAL TIME COUNTER LEAST SIG. WORD
C42	771122	READ 32 BIT REAL TIME COUNTER MOST SIG. WORD
C43	771124	CLEAR MONITOR PENDING INTERRUPT REGISTER
C44	771126	LOAD MONITOR INTERRUPT MASK REGISTER
C45	771130	READ MONITOR PENDING INTERRUPT REGISTER

TABLE 3. (CONTINUED) CONTROL DECODE

DECODE (DECIMAL)	ADDRESS (OCTAL)	FUNCTION
C46	771132	LOAD MONITOR MEMORY #1 512 WORD POINTER
C47	771134	LOAD MONITOR MEMORY #2 512 WORD POINTER
C48	771136	READ MONITOR MEMORY #1 512 WORD POINTER
C49	771140	READ MONITOR MEMORY #2 512 WORD POINTER
C50	771142	LOAD MICROCOMPUTER DMA ADDRESS REGISTER
C51	771144	LOAD MICROCOMPUTER DMA DATA REGISTER
C52	771146	LOAD SIMULATION DISCRETE WORD
C53	771150	READ SIMULATION ADDRESS REGISTER
C54	771152	LOAD SIMULATION DATA REGISTER
C55	771154	READ SIMULATION DATA REGISTER
C56	771156	RESET SIMULATION ACCESS
C57	771160	READ MICROCOMPUTER PC
C58	771162	LOAD TRACE RECORDING SELECTION (TE#1)
C59	771164	LOAD TRACE REGISTER MODE SELECTION (TE#2)
C60	771166	LOAD TRACE REGISTER NUMBERS (TE#3)
C61	771170	READ BIT MAP DATA
C62	771172	READ PANEL OPERAND REGISTER MOST SIG.
C63	771174	READ PANEL OPERAND REGISTER NEXT MOST SIG.
C64	771176	READ PANEL OPERAND REGISTER LEAST SIG.

Master Mode - The Monitor only enters the master mode in order to process monitor generated interrupts and in order to download data from the Monitor Trace memories to the PDP11 main memory. When generating the level five interrupt vector address 120 (octal) is asserted on the data lines. Download data transfer from the Monitor to the PDP11 main memory is in blocks of sixteen words starting with the lowest address in monitor memory transferring to the lowest in page address in PDP11 main memory. The address provided to the PDP11 is the appropriate page register concatenated with the time stamp flag, monitor memory pointer, data/label flag, and the least significant address position (set to zero). See Figure 7, Address Construction.

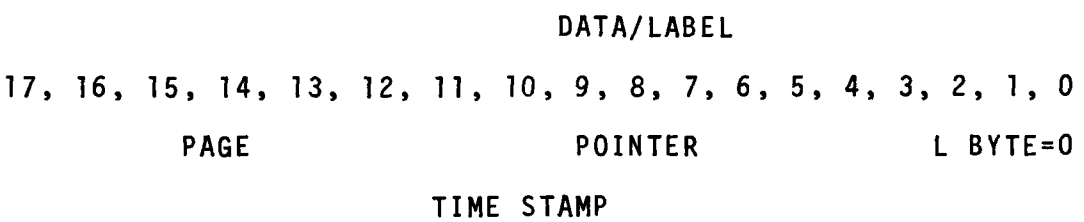


FIGURE 7. ADDRESS CONSTRUCTION

Unibus Timing - Monitor internally generated bus control signals are synchronous with the monitor ten megahertz clock, CL100.

Minimum Acquisition Time	200 Nanoseconds
Minimum Release Time	200 Nanoseconds
Minimum Hardware Interrupt Cycle	500 Nanoseconds
Minimum Slave Response	300 Nanoseconds
Minimum Busy Time for 16 Work Block	8100 Nanoseconds

Direct Memory Access (DMA) - The Monitor has a direct memory access channel with the microcomputer. The channel has the first priority greater than the microcomputer central processing unit. The channel is capable of addressing the 65536 words of microcomputer memory and reading their contents. The channel is capable of writing into any writable location in microcomputer memory. The PDP11 is capable of reading and writing the monitor DMA address and data registers and initiating a DMA transfer with the microcomputer.

Signal Interface - The direct memory access function interfaces with the Unibus receiving data, address, and control discretes from the PDP11 on the D bus. Data is returned from the monitor DMA data register via the BFD bus. The DMA function interfaces with the microcomputer Ibus via the monitor for data and address transmission to the microcomputer and through the UCD bus for data reception from the microcomputer. See Table 4, DMA Signals.

The monitor transmits microcomputer bus request #1 to request the use of the microcomputer bus.

DMA Implementation - See Figure 8, DMA Functional. The channel is implemented as (1) an address register to hold the desired microcomputer address, (2) a data register and multiplexer capable of receiving data from the PDP11 or the microcomputer, and (3) interface and control logic to the BFD, I, D, and UCD busses.

The channel initiates a transfer with the microcomputer by the PDP11 action of loading the DMA address register.

DMA Timing - The generation of each individual read of write transaction after receipt of the DMA address consumes no more than one microsecond if the microprocessor bus is available.

Panel Instruction Facility - The Monitor provides the facility to simulate CPU memory references for instructions and (by User selection) to allow data to be transmitted or received by memory or the monitor. Instruction execution of the panel instruction may be repetitive or single stepped.

Signal Interface - See Table 5. The monitor panel instruction facility interfaces with the PDP11 to receive instruction values, receive/transmit operand values, and receive control discretes. The facility interfaces with the Ibus, substituting for the memory in CPU transactions providing data and control.

Panel Instruction Implementation - See Figure 9, Panel Instruction Functional. The Panel Instruction Facility is capable of simulating memory by providing any combination of instruction data and operand data selected. The implementation is composed of the necessary registers and control logic to interface with the Unibus and microcomputer Ibus and transmit or receive the User selected data.

TABLE 4. DMA SIGNALS

DESCRIPTION	SYMBOL
DMA Data Load Pulse	$\overline{C51}$
DMA Address Load Pulse	$\overline{C50}$
Debug Load Pulse (BG)	$\overline{C27}$
DMA Enable Discrete	BG4
DMA Read/Write Selection	BG6
Read DMA Data	$\overline{C25}$
Read DMA Address	$\overline{C24}$
Microcomputer Address 0:15	MAL
Microcomputer Data 0:15	IBUS
Microcomputer Bus Grant #1	$\overline{UBG1}$
Microcomputer Acknowledge	$\overline{ACK}$
Microcomputer Data Available	$\overline{DA}$
Microcomputer Data Release	$\overline{DR}$
Microcomputer Memory Transfer	$\overline{IO}$
Microcomputer Mode	$\overline{UWRITE}$
Microcomputer Initiate Pulse	$\overline{IP}$
Microcomputer Bus Request #1	$\overline{BRQ1}$

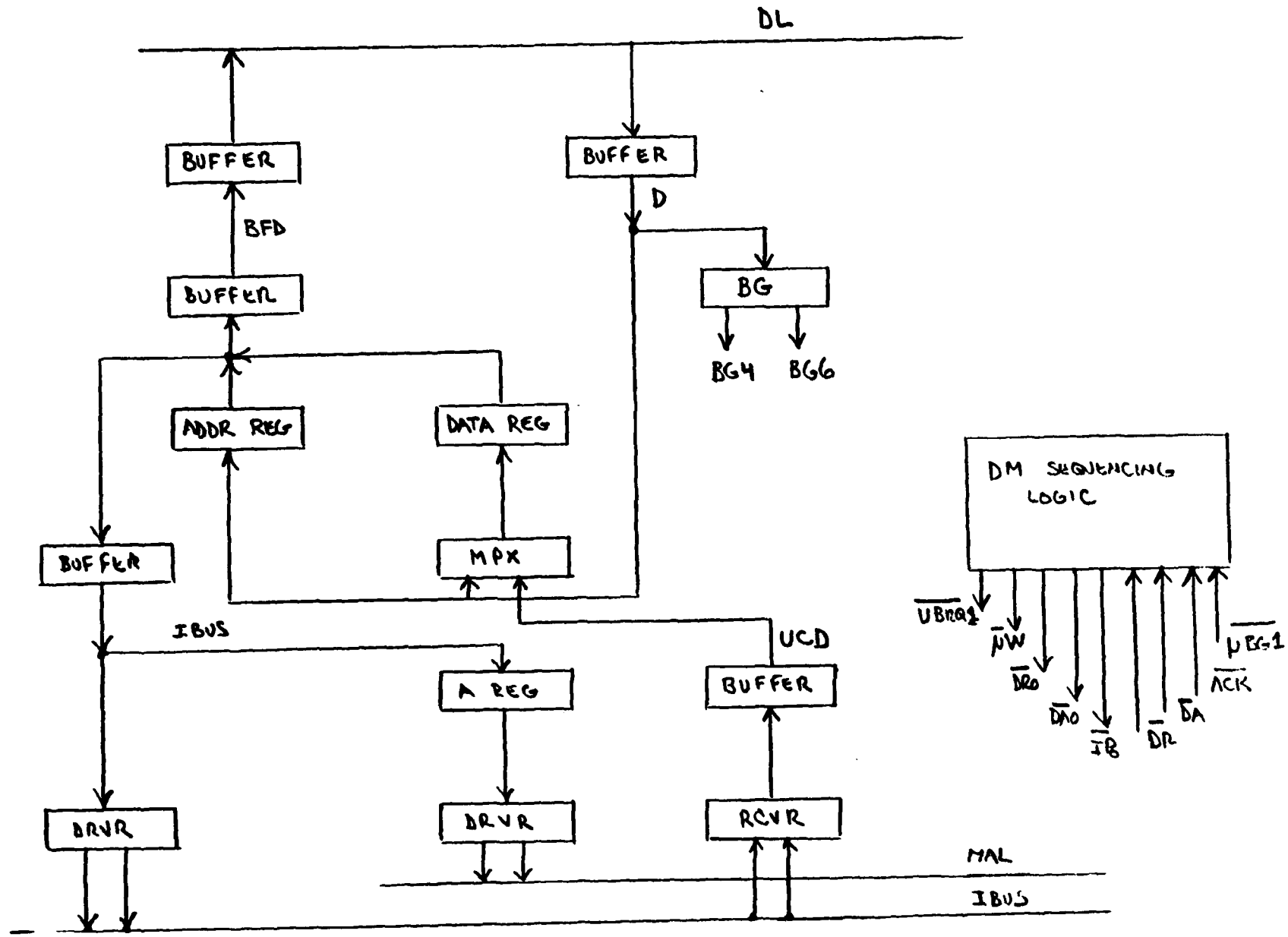


FIGURE 8. DMA FUNCTIONAL

TABLE 5. PANEL INSTRUCTION INTERFACE

DESCRIPTION	SYMBOL
Debug Discrete Load Pulse	$\overline{C27}$
Monitor Microcomputer Command Load Pulse	$\overline{C28}$
Monitor Microcomputer Word #2 Load Pulse	$\overline{C29}$
Monitor Most Sig. OP'D. Register Load Pulse	$\overline{C30}$
Monitor Next Sig. OP'D. Register Load Pulse	$\overline{C31}$
Monitor Least Sig. OP'D. Register Load Pulse	$\overline{C32}$
Monitor Most Sig. OP'D. Register Read Pulse	$\overline{C62}$
Monitor Next Sig. OP'D. Register Read Pulse	$\overline{C63}$
Monitor Least Sig. OP'D. Register Read Pulse	$\overline{C64}$
Debug Discrete Zero	BG0
Debug Discrete One	BG1
Debug Discrete Two	BG2
Debug Discrete Three	BG3
Debug Discrete Five	BG5
Microcomputer CPU Bus Transaction	$\overline{NBG0}$
Initiate Pulse	$\overline{IP}$
Initiate Pulse Acknowledge	$\overline{ACK}$
Data Available	$\overline{DA}$
Data Release	$\overline{DR}$
Microcomputer Memory Inhibit	$\overline{MINH}$
Microcomputer Data	IBUS
Microcomputer Write	$\overline{NWRITE}$

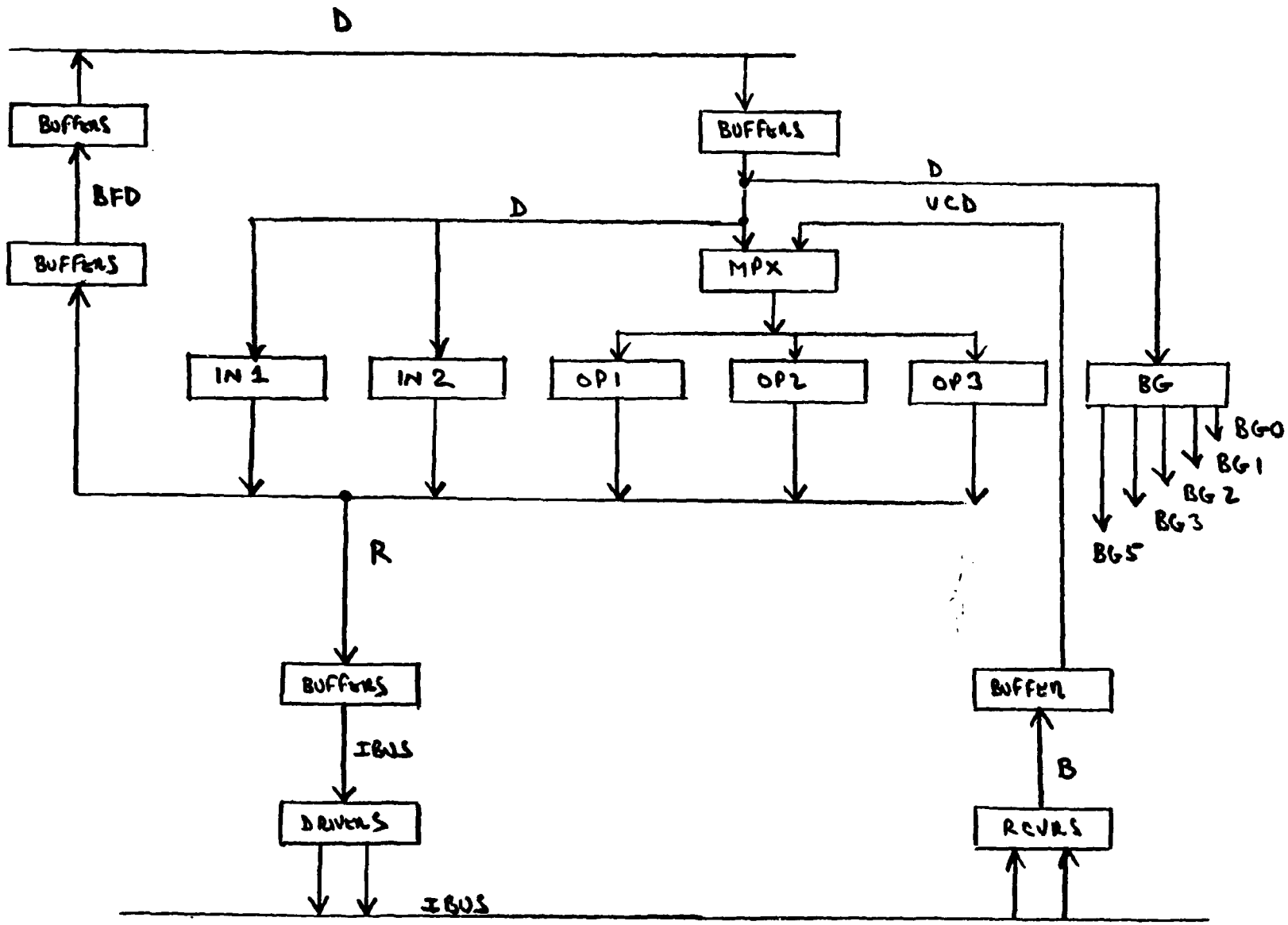


FIGURE 9. PANEL INSTRUCTION FUNCTIONAL

Registers IN1 and IN2 receive the microcomputer command and second instruction word from the PDP11 on the D bus. If instruction simulation is selected these registers provide the instruction to the microcomputer via the R bus and Ibus.

Registers OP1, OP2, and OP3 receive or transmit operand data. OP1 shall always transmit or receive the most significant data word, OP2 the next, and OP3 the least. These registers are loaded from the PDP11 over the D bus or from the UCD on a microcomputer store if operands from microcomputer memory have been User selected.

The debug discrete register, BG, shall receive the control discretes from the PDP11 over the D bus. BG0, BG1, BG2 and BG3 control panel instruction modes.

MODE	BG0	BG1	BG2	BG3
NO SIMULATION	0	0	0	0
INSTRUCTION ONLY	1	①	1	0
OPERAND ONLY (STORE)	0	0	1	0
OPERAND ONLY (LOAD)	0	0	0	1
INSTRUCTION AND OPERAND (STORE)	1	①	1	0
(LOAD)	1	①	0	1

NOTES: ① REQUIRED FOR TWO WORD INSTRUCTIONS

2 BG2=BG3=1 IS INDETERMINATE

Timing - Simulated instruction execution timing shall conform to the Computer Bus Interface.

Input Output Simulation Facility - The Monitor has the capability of simulating microcomputer CPU programmed input output commands by receiving data from the CPU or providing data to the Microcomputer CPU upon detecting an input output instruction. This capability may be invoked independent of whether the physical input output is installed. The PDP11 is notified of the need to service input output simulation via the Monitor interrupt.

Signal Interface - The Monitor interfaces with PDP11 to initialize the input output simulation mode, transmit or receive microcomputer data, transmit the IO simulation address register to the PDP11, and transmit interrupt requests. See Table 6.

Monitor interrupt sources 9 and 10, respectively, inform the PDP11 of the need to service a simulated microcomputer input request or output transmission.

The Microcomputer address lines are captured on input output commands for interpretation by the PDP11. The Microcomputer data lines may be received or driven depending upon the direction of transfer required.

The Ibus time-out inhibit signal is generated by the Monitor to (1) prevent microcomputer bus timeout, and (2) suspend the microcomputer interval timers while the monitor services the simulation interrupt. The MSIM signal in conjunction with the microcomputer address lines is used to deactivate physical input output in the microcomputer.

Simulation Implementation - See Figure 10, IO Simulation Functional. Simulation may be partitioned into deactivation of microcomputer physical input output, detection of microcomputer input output instructions, and transmission and reception of input output data.

The input output simulation logic uses the Monitor DMA capability to logically shutdown physical microcomputer input output. The programming of a DMA write transaction with B67 set shall cause the generation of the input output simulation control signal to the microcomputer.

Detection of the occurrence of a microcomputer input output instruction is enabled by command. The command sets the input output enable which, in turn, enables a timer. The timer counts from the leading edge of each microcomputer initiate pulse while the Ibus signal, $\overline{IO}$, is false. The timer is clocked by a 5MHz square wave generated by the Monitor. Failure of the initiate pulse to become false within 1.6 microseconds causes (1) the generation of either interrupt #9 or interrupt #10 to the PDP11, and (2) generation of the microcomputer Ibus time-out inhibit signal to the microcomputer. Completion of the microcomputer transaction is initiated by detection of command. The command causes the generation of the microcomputer acknowledge signal and enables the balance of the Ibus handshaking.

TABLE 6. IO SIMULATION SIGNALS

DESCRIPTION	SYMBOL
Load Debug Discrete Load Command	$\overline{C27}$
Load IO Simulation Enable	$\overline{C52}$
Read Monitor IO Simulation Address Command	$\overline{C53}$
Load Monitor IO Simulation Data Command	$\overline{C54}$
Read Monitor IO Simulation Data Register	$\overline{C55}$
Acknowledge IO Initiate Pulse	$\overline{C56}$
Microcomputer IO Input Interrupt Source 9	$\overline{INT9}$
Microcomputer IO Output Interrupt Source 10	$\overline{INT10}$
Microcomputer Address Lines	MAL
Microcomputer Data Lines	IBUS
Microcomputer IBUS Timeout Inhibit	BTOINH
Microcomputer IO Simulation Toggle	MSIM
Microcomputer IBUS Grant Zero	BGO
Microcomputer Initiate Pulse	$\overline{IP}$
Microcomputer Initiate Pulse Acknowledge	$\overline{ACK}$
Microcomputer Data Available	$\overline{DA}$
Microcomputer Data Release	$\overline{DR}$
Microcomputer Read	$\overline{W}$
Microcomputer Memory Reference	$\overline{IO}$

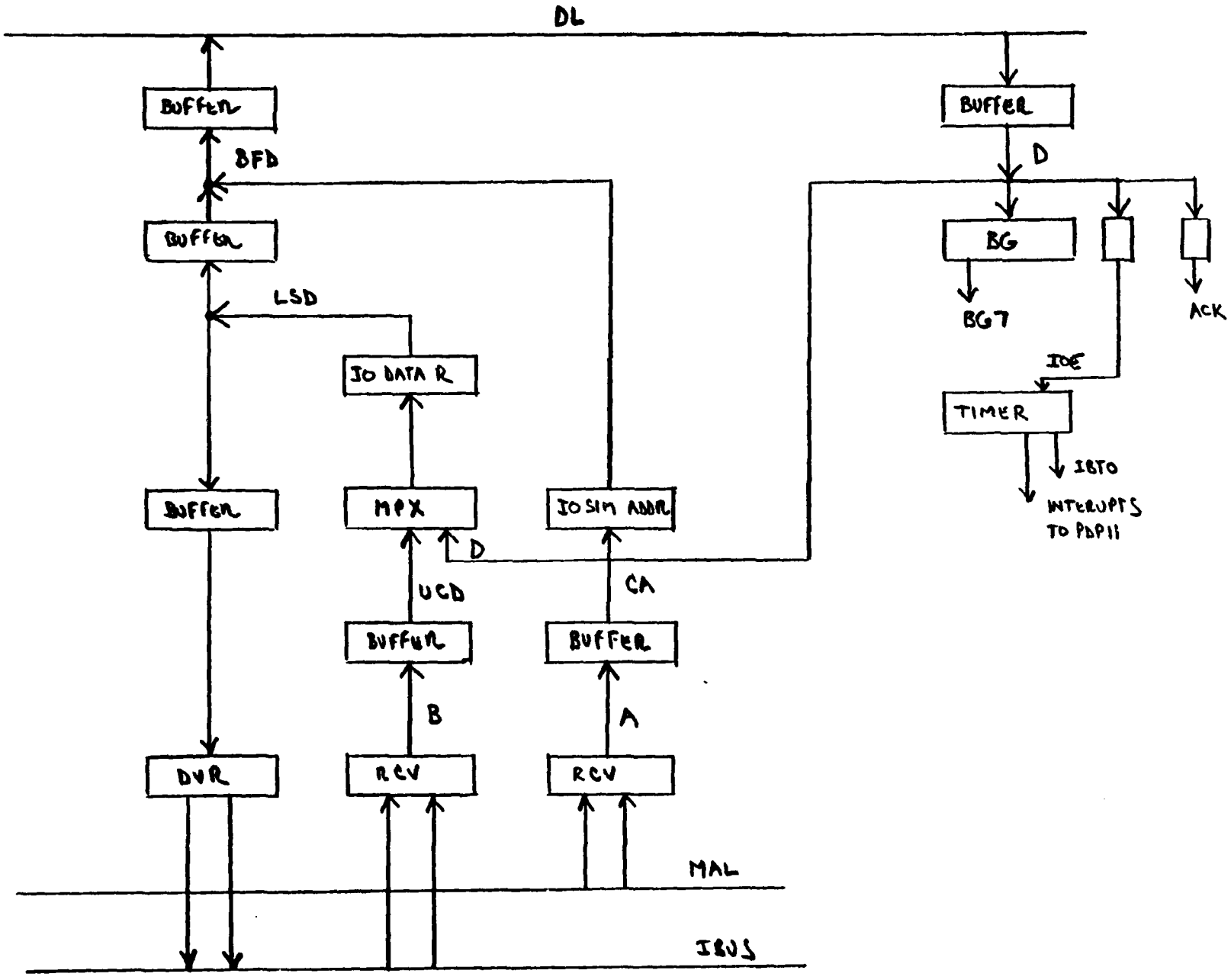


FIGURE 10. IO SIMULATION FUNCTIONAL

Commands with data and data to be input to the microcomputer are loaded into the monitor via the D bus. An IO simulation address register is provided to capture microcomputer input output command. This register and the IO data register are readable over the monitor BFD bus.

Input Output Simulation Timing - Monitor register loads, reads, and commands shall meet the Unibus timing requirements. The pause introduced into the microcomputer Ibus timing following the leading edge of the initiate pulse prior to initiation of the Monitor generated acknowledge is at least 1.6 microseconds. There is no restriction on the maximum length pause.

Stop Facility - The Monitor possesses the capability to stop the Microcomputer under a number of different conditions and inform the PDP11 through the monitor interrupt when the stop has been accomplished. The monitor also possesses the capability to release the stop enabling the microcomputer to resume program execution.

PDP11 hardware generated or monitor software generated monitor initialization causes a stop request. Monitor software may generate an unconditional stop. When tracing, if stop on excess rate has been selected and an excessive input rate is detected, a stop request is generated. All other stop requests require further user selection.

The user is able to select (1) stopping on microcomputer address, (2) stopping on detection of a microcomputer generated breakpoint, (3) after the microcomputer execution of a single instruction, (4) upon detection of a monitor event counter overflow, (5) upon detection of the monitor real time counter overflow, (6) after detection of a user selected Trace condition, or (7) after detection of a register Trace recording condition.

The user is able to select the events to be counted by the monitor event counter. The user selection shall be (1) each occurrence of a microcomputer instruction, (2) each occurrence of a microcomputer instruction in a user specified region, (3) each occurrence of a microcomputer non-CPU bus transaction, (4) each occurrence of a non-CPU bus transaction in a user specified region, (5) each operand read or write, (6) each operand read or write in a user specified region, (7) each user specified instruction type access in a user specified region, (8) each program interrupt, or (9) user specified program interrupt levels.

Signal Interface - (See Table 7.) The monitor stop facility interfaces with the Unibus to receive commands, transmit real time and event counter values, transmit interrupt status, and request program interruption on PDP11 level five.

The monitor stop logic interfaces with the microcomputer to generate clear and stop request signals and receive address, data available, stop acknowledge, and microcomputer microphase label.

The stop logic receives the user selection region flag from the monitor bit map facility.

Unibus initialization is the PDP11 power-up system initialization signal.

Interrupt source 6 and 7 are the microcomputer stop completion notification for a user selected stop other than microcomputer register recording or a stop for register recording, respectively.

The microcomputer address lines are used for stop on address comparison and for Bit Map Facility accessing. The user selected memory region is flagged by the signal BME from the bit map.

The microcomputer data available pulse identified when valid data is on the microcomputer Ibus. It is used in the stop logic to generate the event counter clock.

Logic reset is transmitted to the microcomputer to initialize its condition. The stop request and microcomputer stop acknowledge signals are generated by the monitor stop logic and microcomputer respectively to synchronize stopping the microcomputer.

The microcomputer microphase label is a four bit field embedded in the microcomputer microcode identifying the current CPU action. This field is continuously transmitted to the monitor.

Stop Facility Implementation - See Figure 11, Stop Functional. The monitor stop facility shall latch stop requests from the following sources when appropriate:

a. Initialization
b. Unconditional Stop Request
c. Excess Trace Input Rate
d. Microcomputer Breakpoint Recognition
e. Register Stop Request
f. Single Step Execution
g. Stop On Address Condition Met
h. Event Counter Overflow
i. Real Time Counter Overflow
j. Trace Condition Met

TABLE 7. STOP FUNCTION SIGNALS

DESCRIPTION	SYMBOL
Unconditional Stop Command	$\overline{C1}$
Monitor Reset Command	$\overline{C40}$
Microcomputer Reset Command	$\overline{C26}$
Microcomputer Start Command	$\overline{C12}$
Stop Discrete Word #1 Load Pulse	$\overline{C2}$
Stop Discrete Word #2 Load Pulse	$\overline{C3}$
Debug Discrete Load Pulse	$\overline{C27}$
Counter Timer Discrete Load Pulse	$\overline{C7}$
Event Counter Load Pulses	$\overline{C4}$, $\overline{C5}$
Real Time Counter Load Pulses	$\overline{C9}$, $\overline{C10}$
Event Counter Read Pulses	$\overline{C36}$, $\overline{C37}$
Real Time Counter Read Pulses	$\overline{C41}$, $\overline{C42}$
Unibus Initialization	INITL
Monitor Interrupt Source 6	$\overline{INTERRUPT6}$
Monitor Interrupt Source 7	$\overline{INTERRUPT7}$
Microcomputer Address Lines	$\overline{MAL}$
User Selected Memory Region	$\overline{DME}$
Microcomputer Data Available	$\overline{DA}$
Logic Reset to Microcomputer (Clear)	$\overline{LR}$
Stop Request to Microcomputer	$\overline{STOP}$
Microcomputer Stop Acknowledge	$\overline{IDLE}$
Microcomputer Microphase Label	MM44...MM47

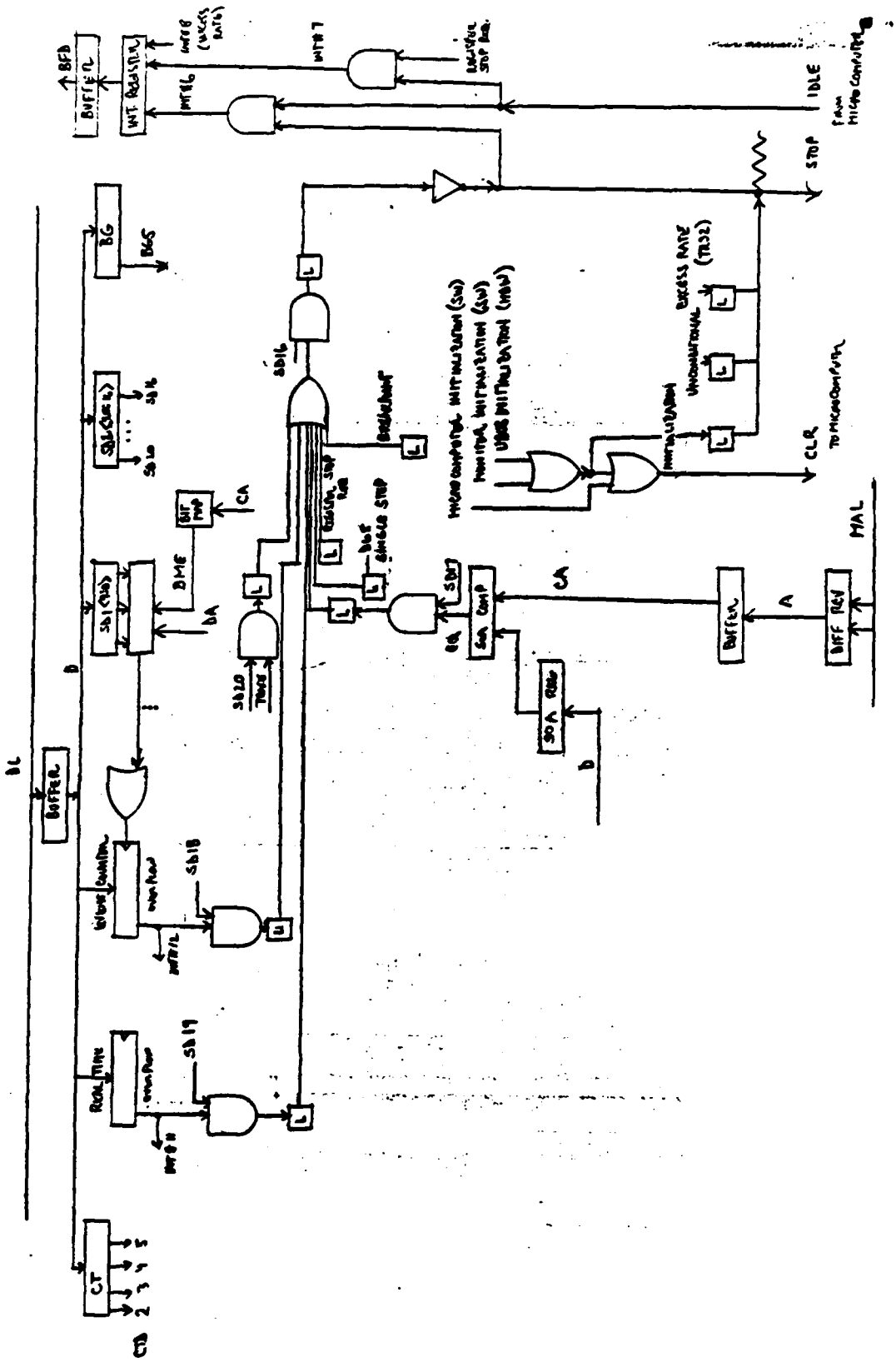


FIGURE 11. STOP FUNCTIONAL

Generation of the latched output of initialization, unconditional stop, or excess Trace rate is sufficient to generate a stop request to the microcomputer. The balance of the sources require additional stop conditions. The microcomputer generates the idle input to the monitor when the CPU ceases normal instruction execution.

The monitor generates the following stop related program interrupt inputs to the monitor pending interrupt register when the microcomputer stops:

a. Interrupt #6 for any stop condition met
b. Interrupt #7 for a stop for microcomputer register transfer
c. Interrupt #8 for a stop due to excessive Trace input rate.

In addition, Interrupt #11 is generated when the real time counter overflows and Interrupt #12 is generated when the event counter overflows.

All latched stop requests are reset by the microcomputer run command, with the exception of the excess rate stop and the register stop which are removed by detection of an available recording buffer and completion of a register set recording, respectively.

Real Time Implementation - Real time counting is implemented by providing a 32 bit up counter driven by a user selectable monitor clock input frequency of either one megahertz or 31.25 kilohertz. The monitor master oscillator generates 20 megahertz stable to $\pm$1 kilohertz from which the real time input frequencies are produced.

The real time counter is loadable and readable by the user. The real time counter generates interrupt source #11 when full count is reached. The real time counter pauses in count during the microcomputer IDLE state.

The user has the capability to count elapsed microcomputer execution time or elapsed microcomputer execution time for microcomputer programs with instructions in user selected regions. Input frequency of either 1 megahertz or 31.25 kilohertz may be selected.

Bit Map Facility - The monitor possesses a facility that enables a user to select from none to all 65536 microcomputer memory addresses as selected as a condition for stopping, tracing, or counting. The facility does not impede the execution time of the microcomputer.

Bit Map Signal Interface - See Tables 8 and 9. The Bit Map interfaces with the PDP11 for loading, verification, and control. The Map interfaces with the microcomputer address lines for detection of user selected addresses. The Bit Map outputs a marked address flag to using facilities.

Bit Map Facility Implementation - See Figure 12, Bit Map Functional. The Bit Map is implemented as a writable random access memory. Each individual bit in the random access memory corresponds to a user selected microcomputer memory address.

Sixteen columns of data are written from the monitor data register. Data from all sixteen columns is read to the PDP11. Only the appropriate single column and address are active when sensing microcomputer addresses.

Timing - Bit map determination of a user selected region does not affect the execution time of the microcomputer.

Trace Facility - The monitor provides a facility for recording user selected microcomputer activity. Concurrent recording and downloading to the PDP11 is provided. The user has the capability of selecting the data to be collected. The monitor records a label identifying each item recorded. The monitor has a user selectable capability to record the time of recording of each data item. The user has the capability of selecting to suspend recording microcomputer data until buffer space is available or stopping microcomputer execution until buffer space is available. The user possesses the capability of suspending downloading. The monitor transfers sixteen word blocks using the non-processor transfer capability of the PDP11 Unibus. The monitor generates interrupt requests to the monitor interrupt facility (1) when a buffer download has been completed, or (2) when an excessive input rate has been detected, or (3) when a stop request for microcomputer register transfer has been honored.

Trace Signal Interface - The Trace logic may be conveniently divided into recording logic and downloading logic. The recording interface is concerned with establishing user selected recording conditions, identifying their occurrence from microcomputer signals, and writing into the monitor buffer memories. The download interface is concerning with detected full buffers and unloading the buffer contents to the PDP11 through the Unibus.

The recording interface, See Table 10 Recording Signals, utilizes commands to load the user selected tracing conditions. The buffer address registers are loaded and read by commands for registers one and two, respectively. The microcomputer data and address busses are routed to the MC data buffer memories for

TABLE 8. BIT MAP INTERFACE SIGNALS

DESCRIPTION	SYMBOL
Bit Map Data Register Load Command	C33
Bit Map Address Register Load Command	C34
Bit Map Control Flags Load Command	C35
Bit Map PDP11 Read Command	C61
Bit Map Write Flag	FWRITE
Bit Map Monitor Address Source Flag	OE1
Bit Map Data Output Marked Address	BME
Microcomputer Address Lines	MAL

TABLE 9. MAP CONTROL

D1 FWRITE	D0 OE1	DESCRIPTION
0	0	Sample Microcomputer
1	1	Monitor Write Map
0	1	PDP11 Read Map
1	0	Undefined

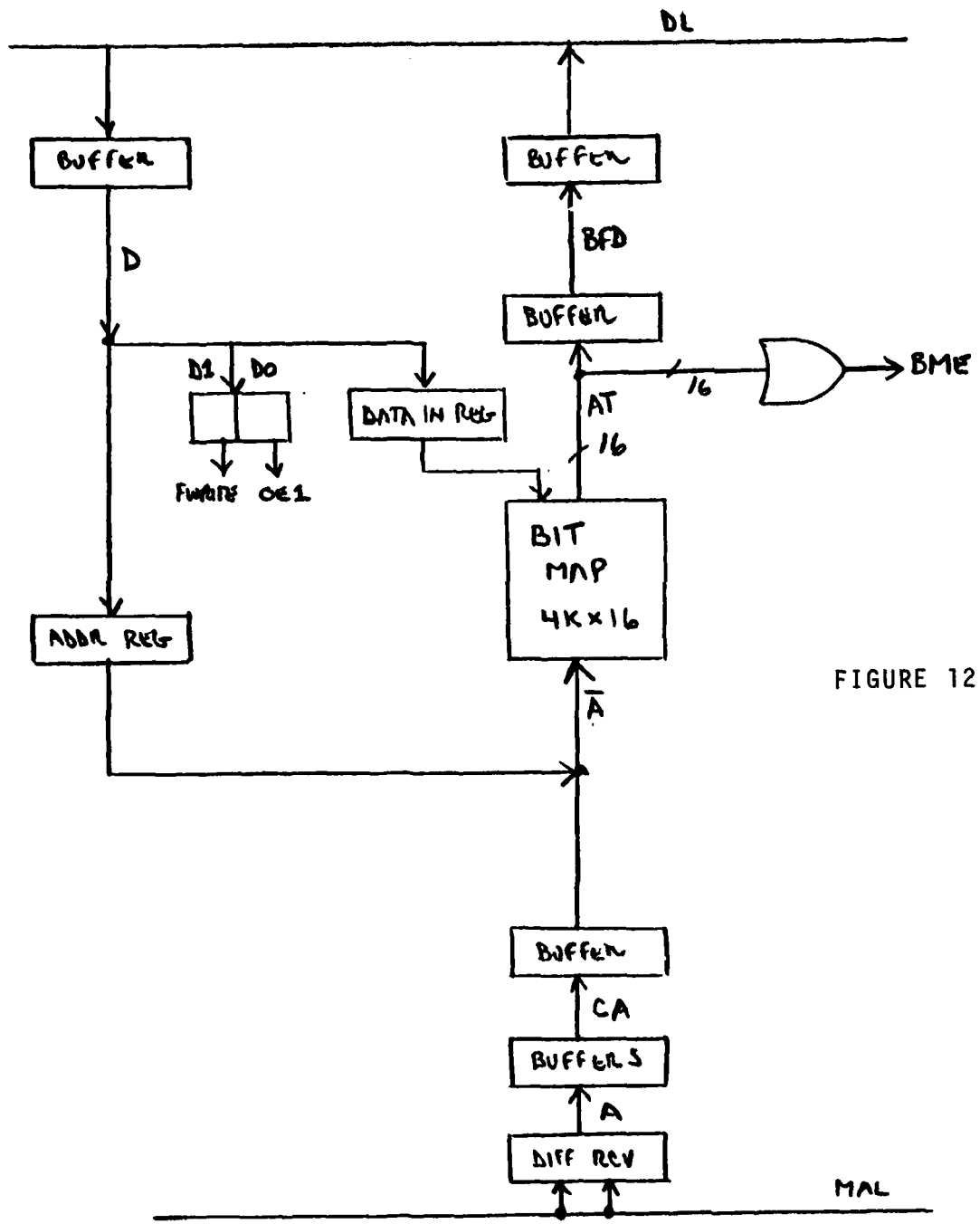


FIGURE 12. BIT MAP FUNCTIONAL

TABLE 10. RECORDING SIGNALS

DESCRIPTION	SYMBOL
Bus Recording Discrete Word #1, Load Pulse, TE#1	C58
Register Recording Discrete Word #2, Load Pulse, TE#1	C59
Register Selection Discrete Word #3, Load Pulse, TE#3	C60
Path Trace Discrete Word #1, Load Pulse, TR#1	C14
Value Trace Discrete Word #2, Load Pulse, TR#2	C15
Misc. Discrete Word #3, Load Pulse, TR#3	C16
Pointer 1 Load Pulse	C46
Read Pulse	C48
Pointer 2 Load Pulse	C47
Read Pulse	C49
Microcomputer IBUS 0:15	IBUS
Microcomputer Address Lines 0:15	MAL
Idle Line	IDLE
Data Release	DR
Data Available	DA
Register Codes 3:0	RC
Microcomputer Microphase 44:47	MM

recording. The microcomputer microphase label, the microcomputer write signal, the register codes, and information derived from the user recording selections are routed to the monitor label memories. The monitor time stamp memories are loaded from the least significant portion of the real time counter.

The download interface, see Table 11 Download Signals, controls the downloading of Trace data from the monitor buffer memories to the PDP11. Buffer Flags #1 and #2 are software set flags enabling the monitor to download the associated buffer when it becomes full. The flags are reset by the hardware when the associated buffer download has been completed. Commands load page registers 1 and 2 respectively and load Flags #1 and #2 concurrently. Interrupt sources #8, #15 and #14 are sources to the monitor interrupt logic to notify the PDP11 of an excess input rate, completion of buffer download #1, and completion of buffer download #2, respectively.

Trace Implementation - See Figures 13 and 14, Trace Functional. The Trace facility is implemented as two sets of 512 word semiconductor memories. Each set is addressed by its own address register. Each set is composed of three 16 bit sections dedicated to microcomputer information, label, and time recording and downloading.

Concurrent recording in one memory set while downloading from the other memory set is implemented. Recording commences with memory set #1. Downloading to the PDP11 is under non-processor transfer protocol in blocks of sixteen words.

Recording within a memory set is concurrent for the data, label, and time sections. Downloading is section sequential with 512 words of data followed by 512 words of label, followed by 512 words of time if the time stamp option has been selected.

The downloading address to the PDP11 is composed of the concatenation of a logic zero, the appropriate monitor memory address register output, a data/label toggle, a time stamp toggle, the appropriate monitor memory page register output, concluding with two logic zeroes for the most significant addresses (A16, A17). During recording and downloading the monitor memory address registers advance from zero to 511 as either process progresses. The occurrence of 511 shall indicate a fully recorded buffer memory or the conclusion of a memory section download.

Initiation of downloading to the PDP11 is prevented until the user sets the BLT flag for the associated buffer memory. Completion of a buffer memory transfer generates monitor source interrupt #15 for monitor buffer #1 or monitor source interrupt #14 for monitor buffer #2. If the recording rate causes a monitor buffer memory

TABLE 11. DOWNLOAD SIGNALS

DESCRIPTION	SYMBOL
Buffer Flag 1	BLT #1
Buffer Flag 2	BLT #2
Page #1 Load Pulse	C38
Page #2 Load Pulse	C39
Non-Processor Request	NPR
Non-Processor Grant	NPG
Bus Busy	BBSY
Master Synch	MSYN
Slave Synch	SSYN
Mode Lines	C0, C1
Monitor Output Data Bus to PDP11	BFD
Monitor Output Address Bus to PDP11	ARH
Excess Input Rate Interrupt	INT #8
Monitor Buffer Memory #1 Download Complete	INT #15
Monitor Buffer Memory #2 Download Complete	INT #14

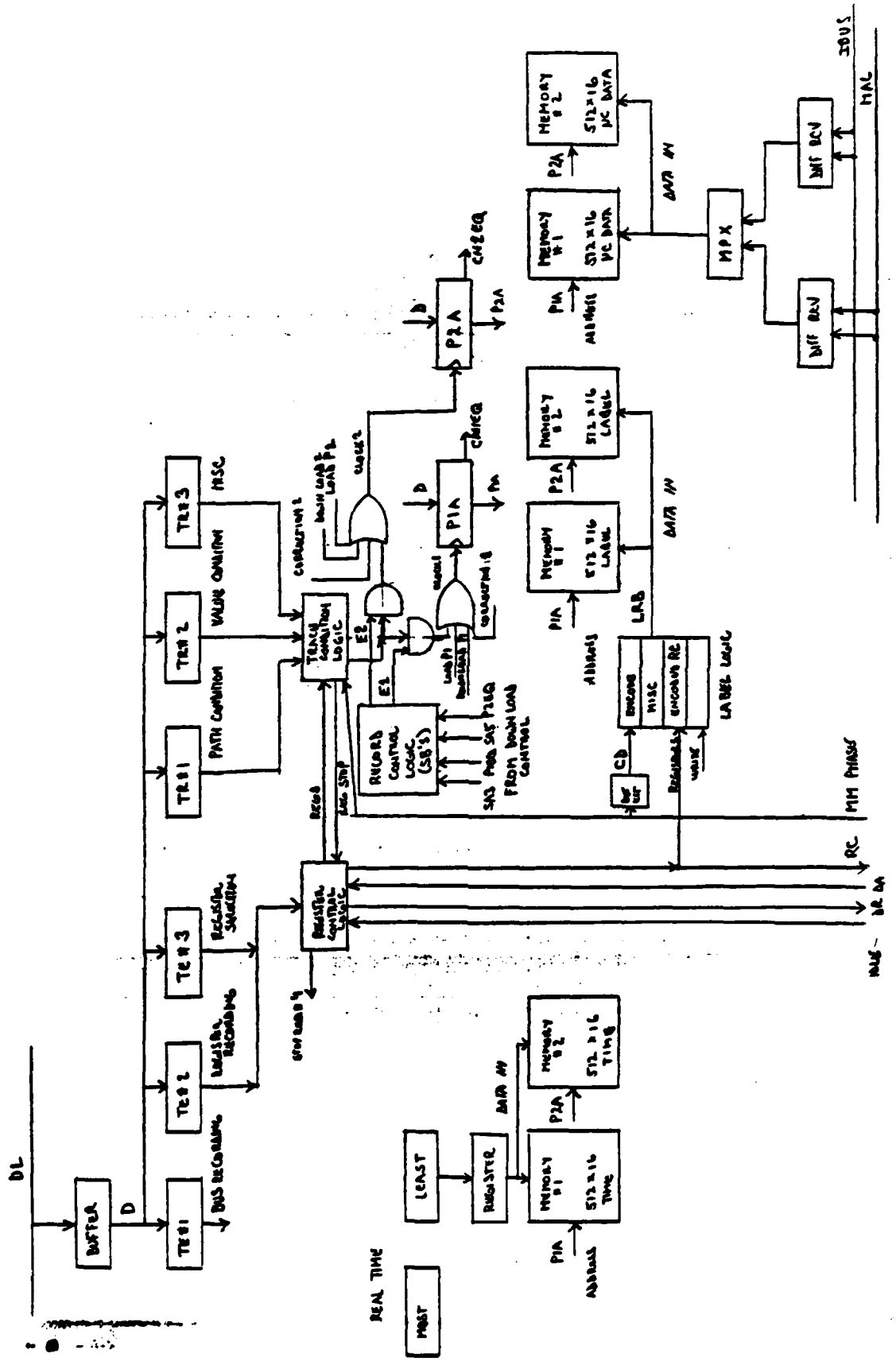


FIGURE 13. TRACE FUNCTIONAL

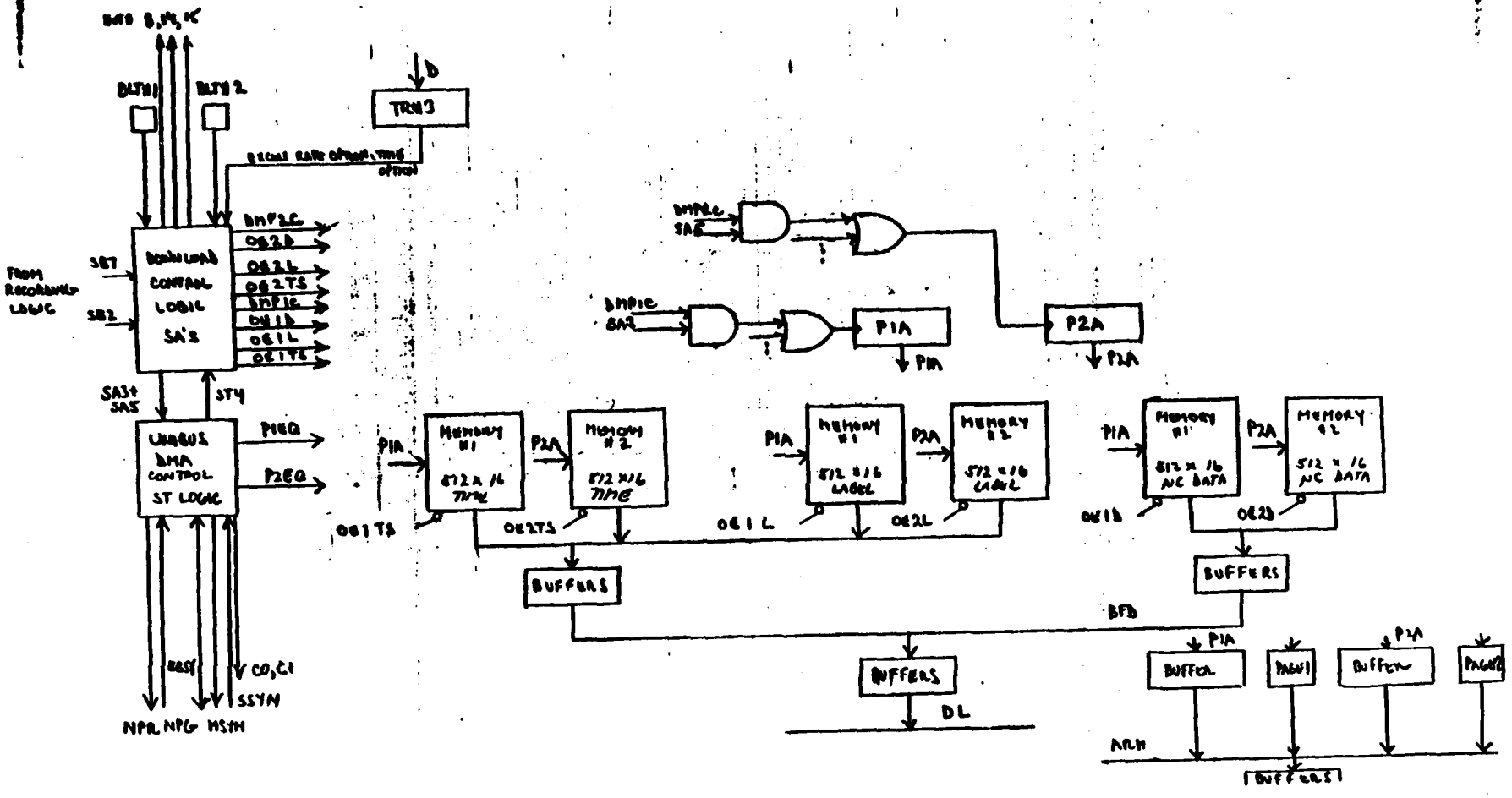


FIGURE 14. TRACE FUNCTIONAL

to be required while that memory is being downloaded from a prior recording the excess rate monitor interrupt source #8 is generated. By prior use selection either the microcomputer is stopped until a monitor recording buffer is available or recording is skipped until a monitor buffer is available.

When Trace recording has been selected, write enable pulses are generated after the leading edge of the microcomputer initiate pulse, IP, and after the leading edge of the microcomputer data available pulse, respectively, if both address and data have been selected for tracing. Otherwise, the write enable pulse is only generated with the user selection of address or data exclusively.

Following each recording the associated monitor buffer address register is advanced if the Trace condition logic detects the data as associated with condition selected by the user for tracing, otherwise the address register may (1) remain unclocked, (2) decremented, or (3) preset to a prior value depending on the user conditions being traced.

Trace enable words determine what microcomputer item(s) are recorded.

The recording selection logic generates register codes, RC0 to 3, to the microcomputer to identify the registers to be placed on the microcomputer Ibus for transmission. If sixteen registers are selected then register codes 0 through 15 are generated in sequence. The microcomputer transfers one register per write cycle while in the Idle state.

The monitor buffer memory Trace condition logic and correction logic provide an input to the clocking logic for the address register of whichever buffer memory is recording. Recording buffer memory selection is accomplished by the record control logic (the SB's).

Trace specification words provide user selection of the conditions to be recorded. This static selection is combined with the microcomputer microphase label, and the microcomputer Ibus control signals to provide buffer address clocking.

Flag zero, TR0, selects tracing specific instruction types in a specific region in memory. This is implemented by detecting instruction fetches with BME true. If a preset value compares with the current microcomputer instruction the recorded instruction is retained, otherwise the address pointer is decremented.

Flag one, TR1, selects tracing all instructions executed in a selected region. Each instruction word detected increments the address pointer.

Flag two, TR2, selects all instructions whose operands are in a selected region. Each instruction execution is recorded if the operand is in the selected region.

Flag three, TR3, selects recording of all operands associated with instructions in a preselected region. The logic shall latch detection of each instruction in the region allowing subsequent recording of the associated operand words.

Flags four, five and six concurrently select three modes of jump tracing as follows:

Flag 4 5 6 Jumps into a Region

Flag 4 $\overline{5}$ 6 Jumps out of a Region

Flag 4 5 $\overline{6}$ All jumps

Jumps into and out of regions are implemented by recording all potentially successful conditions and decrementing the address register if failure is then ascertained.

Flags seven and eight, TR7, TR8, record specific program interrupts (designated by BME) and all program interrupts respectfully.

Flag nine, TR9, selects the recording of non-CPU microcomputer Ibus transactions. This mode shall be implemented by detecting bus transactions when the microcomputer CPU bus grant is false.

Flags ten and eleven, TR10, TR11, detect operand reads and operand writes in a selected region, respectively by detecting concurrence of an operand microphase, BME, and the condition of the read signal.

Trace discrete word #2 selects Trace conditions dependent on data value comparisons with preset limits and traces of instruction addresses and either 16 or 32 bits of instruction value. The functions are implemented utilizing comparison logic and resetting the appropriate buffer memory address register to one more than its last successful recording upon failure. The associated flags shall be set from PDP11 data bits 7 through 15 (TR23 through TR31).

Trace flag twenty-three, (TR23), enable the recording of extended precision floating point data (48 bits) when a comparison of the first thirty-two bits is within the limit register settings.

Trace flags Twenty-four and twenty-five (TR24, TR25), select in range or out of range tracing respectively for 16 bit fixed point data. In range is defined to be the closed interval including the upper and lower limits.

Trace flags twenty-six and twenty-seven (TR26, TR27), select the tracing of thirty-two bit fixed point data in range or out of range respectively. The comparisons are valid for microcomputer store operations only (most significant operand word first in sequence).

Trace flags twenty-eight and twenty-nine (TR28, TR29), select the comparison of the first thirty-two bits of floating point numbers for in range or out of range. Forty-eight (48) bits are recorded for extended precision floating point data if TR23 is true. The comparisons are valid for microcomputer store operations only.

Trace flags thirty and thirty-one (TR30, TR31), select the recording of instruction address and value.

The recording control logic, sequences from buffer memory one to buffer memory two alternating thereafter. The action to be taken, should recording require the use of a buffer memory that has not completed prior downloading, is determined by Trace specification. Flag thirty-two (TR32) enables the initiation of a stop request to the microcomputer. Upon release of the tracing buffer, recording proceeds. Flag thirty-three (TR33), causes recording to be suspended until a buffer is available.

Trace Timing - Trace recording requires no more than three hundred nanoseconds from the leading edge of IP to the leading edge of DA and no more than one hundred nanoseconds from the trailing edge of IP to the leading edge of DA or from the trailing edge of DA to the leading edge of IP.

Contention free downloading requires no more than 8.5 microseconds for each block of sixteen words transferred from the monitor to the PDP11.

Program Interrupt Facility - The monitor is the third ranking member of PDP11 level five for program interruption. The monitor vectors to 12Ø (Octal). The monitor accommodates ten equally ranked interrupt sources latched in a pending interrupt register. Each register source bit is capable of individual program controlled clearing. Each individual source bit is user mask enabled to allow interrupt generation. The latched register is readable.

Interrupt Signal Interface - See Table 12, Interface Signals. Commands clear the interrupt pending register and load the mask register.

TABLE 12. INTERRUPT SIGNALS

DESCRIPTION	SYMBOL
Interrupt Register Clear Command	C43
Mask Register Load Command	C44
Interrupt Register Read Command	C45
Unibus Request	BR5
Unibus Grant	BG5
Unibus Acknowledge	SACK
Unibus Busy	BBSY
Interrupt Request	INTR
CPU Interrupt Acknowledge	SSYN
Microcomputer Stop Acknowledge	INT#6
Microcomputer Register Stop Acknowledge	INT#7
Excessive Trace Rate Warning	INT#8
Simulation Output Request	INT#9
Simulation Input Request	INT#10
32 Bit Timer Overflow	INT#11
32 Bit Counter Overflow	INT#12
16 Bit Timer Overflow	INT#13
Download Memory #2 Complete	INT#14
Download Memory #1 Complete	INT#15

Interrupt #6, Microcomputer Stop Acknowledge, is derived from an input that remains true from when the microcomputer actually stops until the microcomputer resumes.

Interrupt #7, Register Stop Acknowledge, is derived from an input lasting from when the microcomputer stops for a Trace register recording until the microcomputer resumes.

Interrupt #8, Excessive Trace Rate Warning, is derived from a pulse received from the Trace facility.

Interrupts #9 and #10, Simulation Output Request and Simulation Input Requests, are derived from pulses received from the input output simulation facility.

Interrupt #11, Interrupt #12, and Interrupt #13 are derived from pulse inputs indicated that the 32 bit timer, the 32 bit counter and the 16 bit timer, respectively, have overflowed.

Interrupts #14 and #15 are derived from pulses indicating that Trace downloading of monitor buffer memory #2 and a monitor buffer memory #1, respectively, have completed downloading.

Interrupt Facility Implementation - Interrupt sources direct set the individual direct set interrupt register inputs. Generation of an interrupt request to the PDP11 require the logical anding or the equivalent interrupt register and mask register bits. The interrupt service logic sequences through the PDP11 protocol for interrupt generation. The individual interrupt register bits are cleared by presenting a logical one to the appropriate flip flop input.

Interrupt Time - The most rapid contention free monitor portion of interrupt service is 500 nanoseconds.

Microcomputer Detail

The Microcomputer consists of Microprocessor (MC), Memory (M), Programmed Input/Output (PIOU), Direct Memory Access Input Output (DMA), Discrete Input/Output (DIO), Interval Timers (TIO), a MIL-STD-1553B Bus Controller Interface Unit (BCIU), Monitor Interface (MI), and associated power supplies. See Figure 15.

Functional Areas - The major functional areas and interfaces are shown in Figure 16, Microcomputer Functional. The major internal interface between functional areas is the data bus and its associated control signals. The data bus provides a common communications path. Access to the path is on a priority basis arbitrated by logic within the microprocessor. Nominal bus trans-

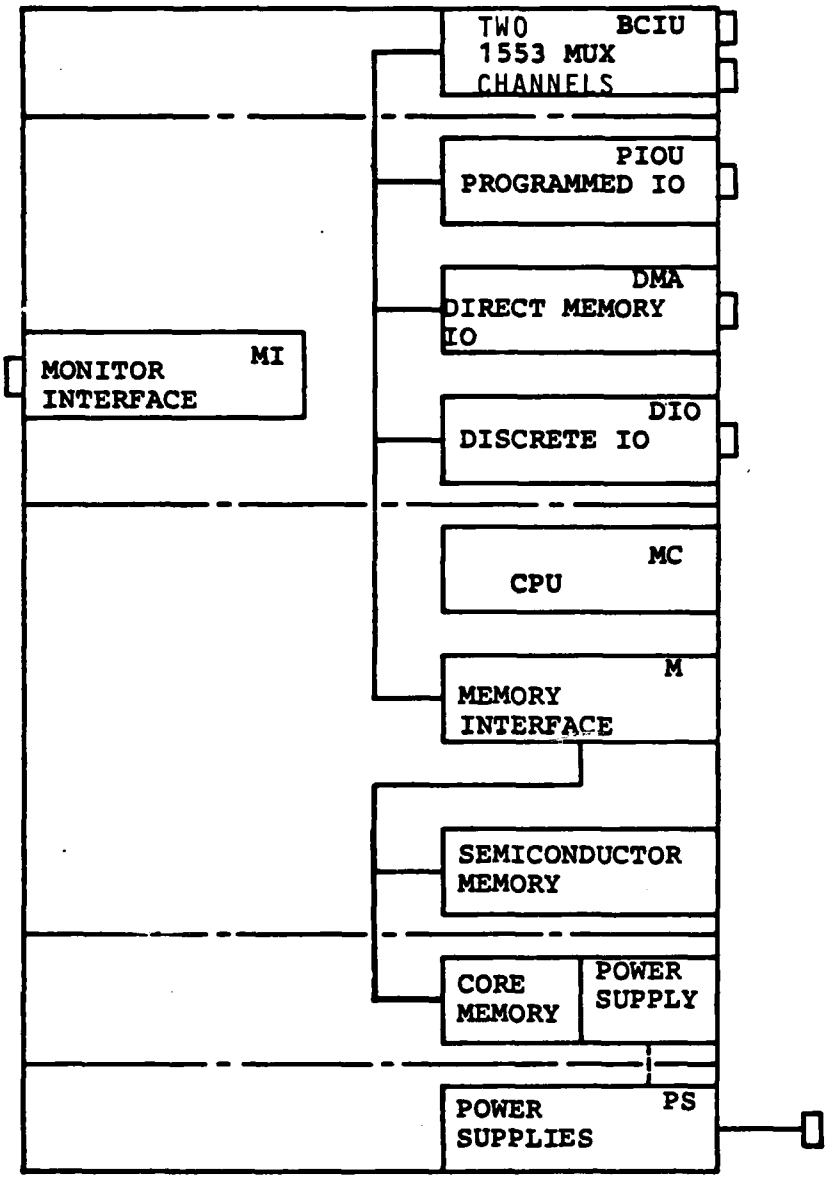


FIGURE 15. MICROCOMPUTER MAJOR AREAS

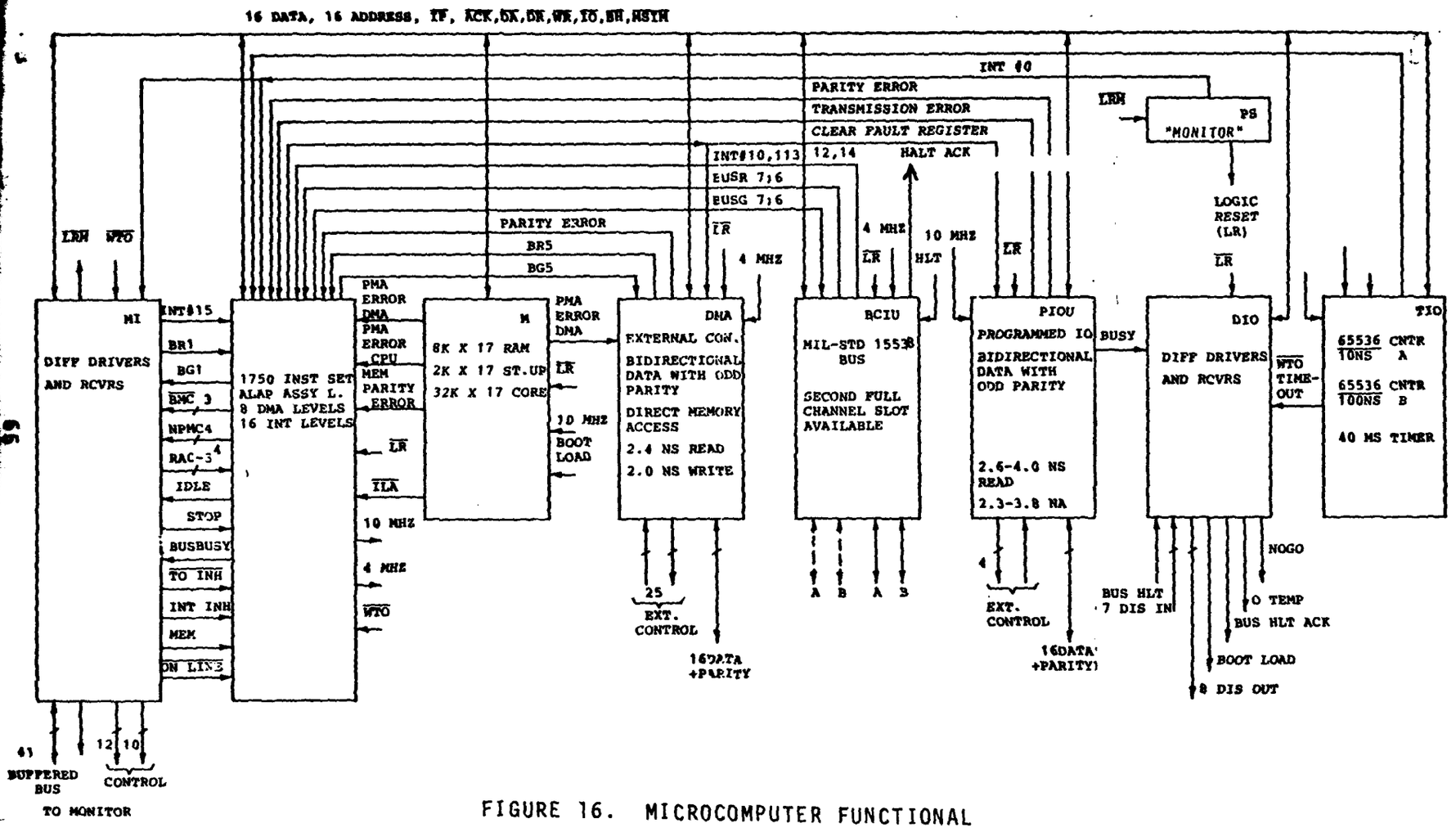


FIGURE 16. MICROCOMPUTER FUNCTIONAL

action time is 0.8 microseconds per 16 bit word.

Microprocessor - The Microprocessor is the computational and control element of the microcomputer operating on input data received from the Direct Memory Access Channel, the Programmed Input/Output, the Bus Controller Interface Unit, and the Discrete Inputs in accordance with instructions from memory. Results are made available through output ports of the sections mentioned. The architecture employed is MIL-STD-1750. External maintenance and software surveillance is accomplished through the Monitor Interface Port.

Memory - The Memory Functional area provides up to 20960 16-bit words of storage for programs and data functions. Sub-functions provided are (1) parity generation of information being written into memory and parity checking of data being read, and (2) separate protected memory features for microprocessor writes and direct memory access writes. The present storage medium is semiconductor and core memory.

Direct Memory Access - The Direct Memory Access Channel provides an externally controlled 16 bit bi-directional data path to memory. Odd parity is implemented on transfers. Read transactions are less than 2.4 microseconds per 16 bit word. Write transactions are less than 2.0 microseconds per 16 bit word.

Bus Controller Interface Unit - The Bus Controller Interface Unit connects a MIL-STD-1553B Bus to the microcomputer, where the BCIU may function in either the master or remote mode. A second controller may be installed for extended application requirements or testing.

Program Controller Input Output Unit - The Program Controlled Input Output Unit allows the microprocessor a 16 bit bi-directional path to external equipment. Odd parity is implemented on data transfers. Read transactions do not exceed 4.0 microseconds per 16 bit word. Write transactions do not exceed 3.8 microseconds per 16 bit word.

Discrete Input Output - The Discrete Input Output area provides 8 latched differential external outputs and 8 external differential inputs. In addition, 5 internal discrete inputs are provided for program sampling.

Timer Input Output - There are two Program Controlled Interval Timers which may be read, written, started, and stopped. Timer A has a 10 microsecond granularity. Timer B has a 100 microsecond granularity. Timer A generates program interrupt #7 on overflow. Timer B generates program interrupt #9 on overflow. A 40 millisecond resettable timer is also provided. If not reset an output discrete is generated.

Power Supplies - The Power Supply function converts AC input power to the internal DC voltages required and provides the power state and logic reset features.

Monitor Interface - The Monitor Interface function provides differential drivers and receivers for the microcomputer bus and for the unique controls between the microcomputer unit and the monitor unit.

Bus Interface - The major internal communications path for all functions is the bus.

Bus access is arbitrated by a controller function implemented in the microprocessor. Eight levels have been assigned as indicated in Table 13. Priority seven is highest.

The Bus Controller includes a bite timer which monitors bus timing and issues an interrupt if a device is on the bus for more than 6.4 microseconds. Bus control is then turned over to the CPU. A block diagram of the Bus Controller is shown in Figure 17.

Microprocessor - The Microprocessor implements the instruction set as defined by MIL-STD-1750. Instruction execution is under microprogram control.

Microprocessor Detail Functional - The MC is composed of:

a. Arithmetic and Logic Unit and Supporting Circuits
b. Microprogram Control
c. Microprogram Memory
d. Address Registers
e. Data/Address Registers
f. Timing and Control Logic
g. Fast Multiply Logic
h. Interrupt Control Logic, and the
i. Bus Controller functions.

These functions are interconnected as shown in Figure 18, Microprocessor Detail Functional.

TABLE 13. BUS PRIORITY

PRIORITY	ASSIGNMENT
7	BCIU #1
6	BCIU #2
5	DMA
4	SPARE
3	SPARE
2	SPARE
1	MONITOR
0	MICROPROCESSOR

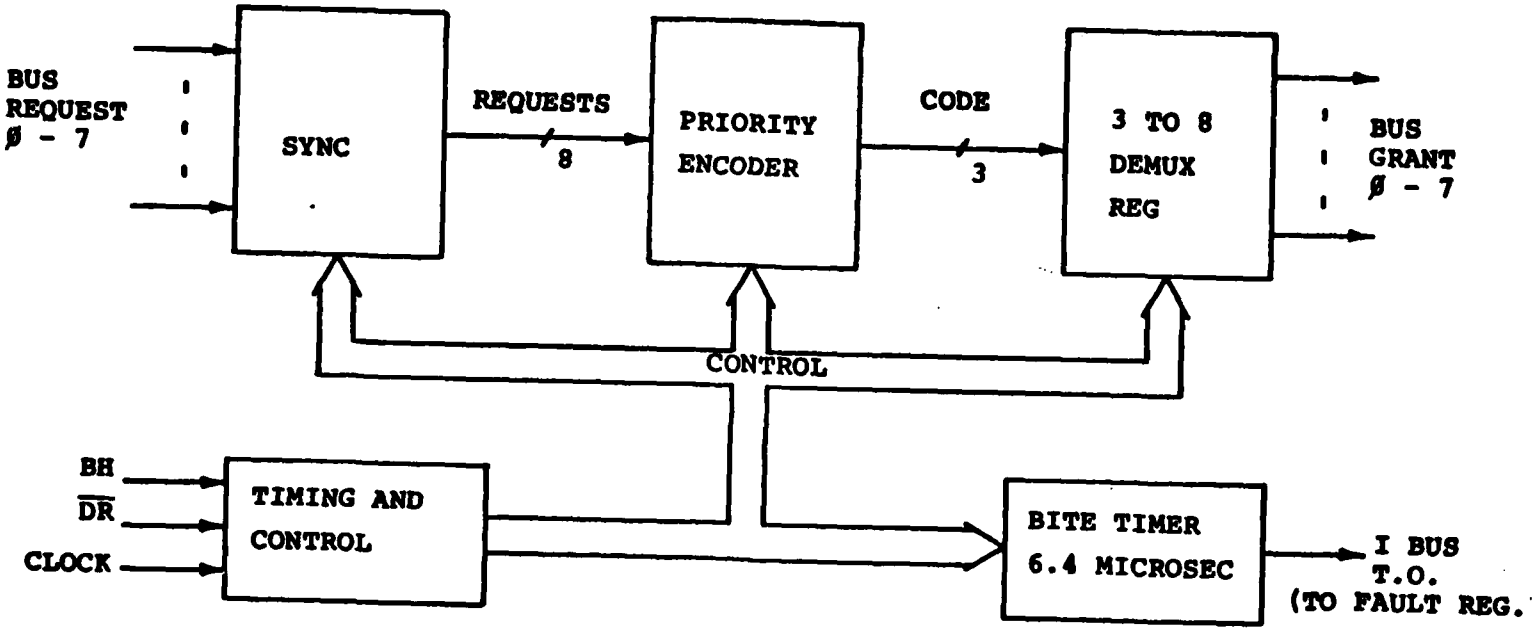


FIGURE 17. BUS CONTROLLER BLOCK DIAGRAM

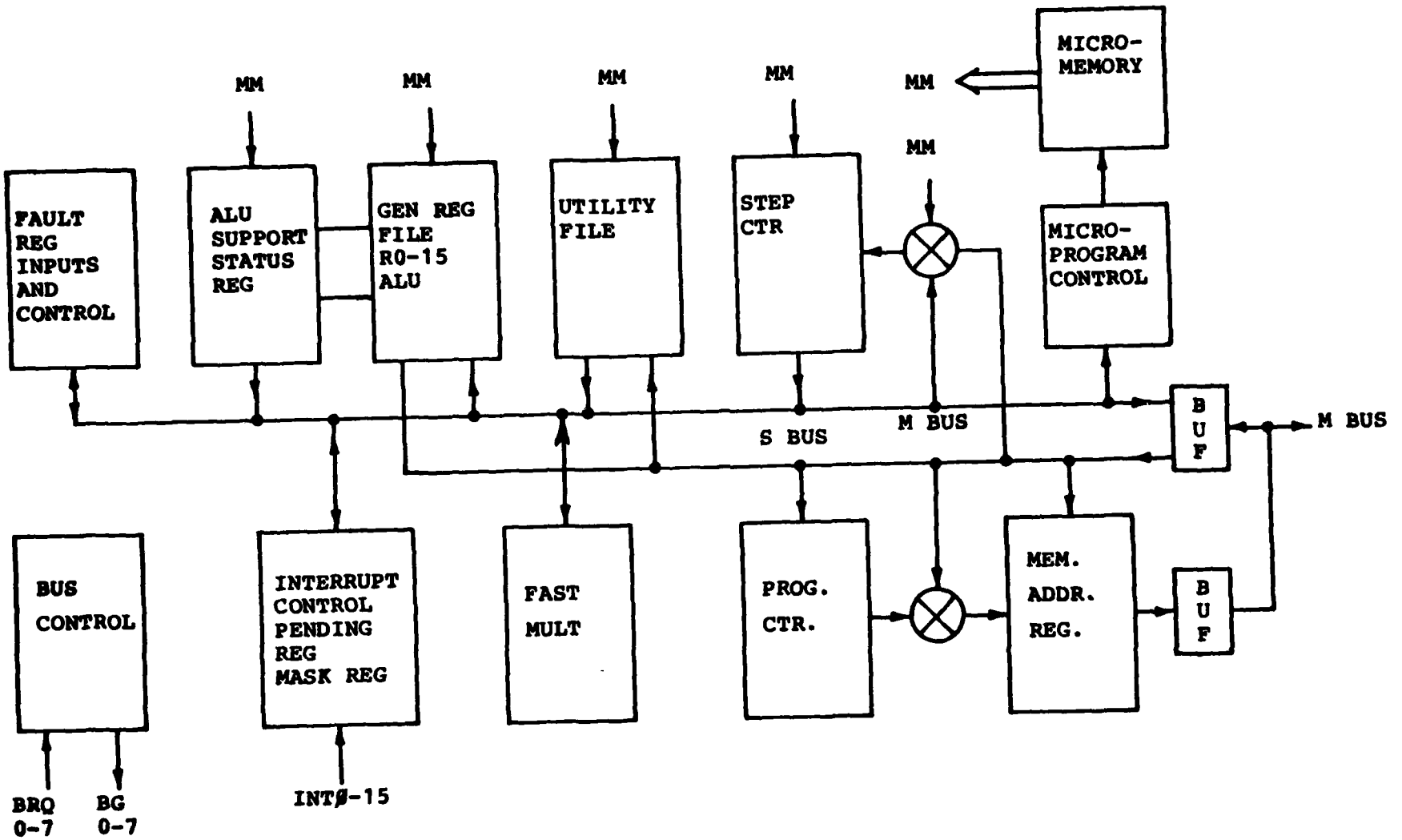


FIGURE 18. MICROPROCESSOR DETAIL FUNCTIONAL

Implementation - The Microcomputer (MC) consists of two cards, CPU #1 and CPU #2. Detailed block diagrams of these cards are shown in Figures 19 and 20, respectively.

Arithmetic and Logic Unit (ALU) - The arithmetic and logical operations in the MC are performed by the AMD 2901 bit-sliced chips, which also contain the 16 general purpose registers accessible to the programmer. Data transfers within the ALU take place via two 16-bit busses, the M-bus and the S-bus. The M-bus is the input to the bit slices and the S-bus is their output; support circuits are connected to one or both of these busses. The following paragraphs provide a brief description of these support circuits.

Register Select Registers - The three registers RA, RB and BR control which of the 16 general registers are to take part in an arithmetic or logical operation. In general, RA selects the register to be operated on, RB the source or index register, and BR the base register.

End Control - This circuitry controls the carry input during an ADD or SUBTRACT operation. During SHIFT operations, it controls the input to the bacated bit positions to allow arithmetic or logical shifting, and single or double precision shifts.

Status Register - The Status Register holds the sign and carry as defined in MIL-STD-1750. It can be read or loaded via the M-bus. It can also be loaded from the result of a bit-slice operation. In order to handle double or triple precision words, the Z (zero) bit is set only if the KS flip-flop is set. Also, the carry bit is set only if the TC field in the microinstruction is two or six; this prevents the carry from being set during logical operations.

External File - This 16 word RAM is used to hold intermediate results during certain instructions.

Step Counter - This 8-bit counter is used to count loops within the microprogram, and as a shift counter during SHIFT instructions.

KS and KC - These two flip flops are used for a variety of purposes depending on the instruction being executed. KS is used to test multiple precision words for ZERO and holds the sign of the divisor during DIVIDE. KC is used for carry propagation in multiple-precision arithmetic.

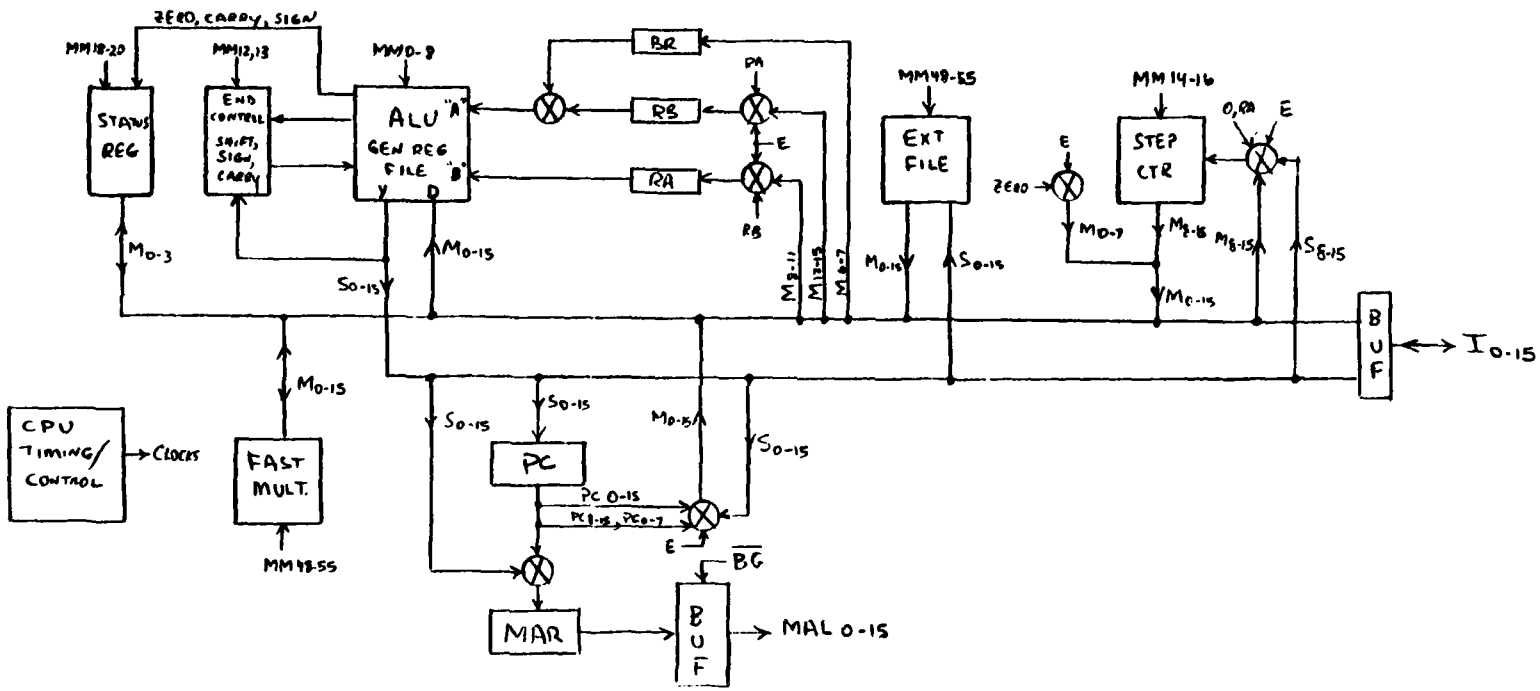


FIGURE 19. CPU #1 BLOCK DIAGRAM

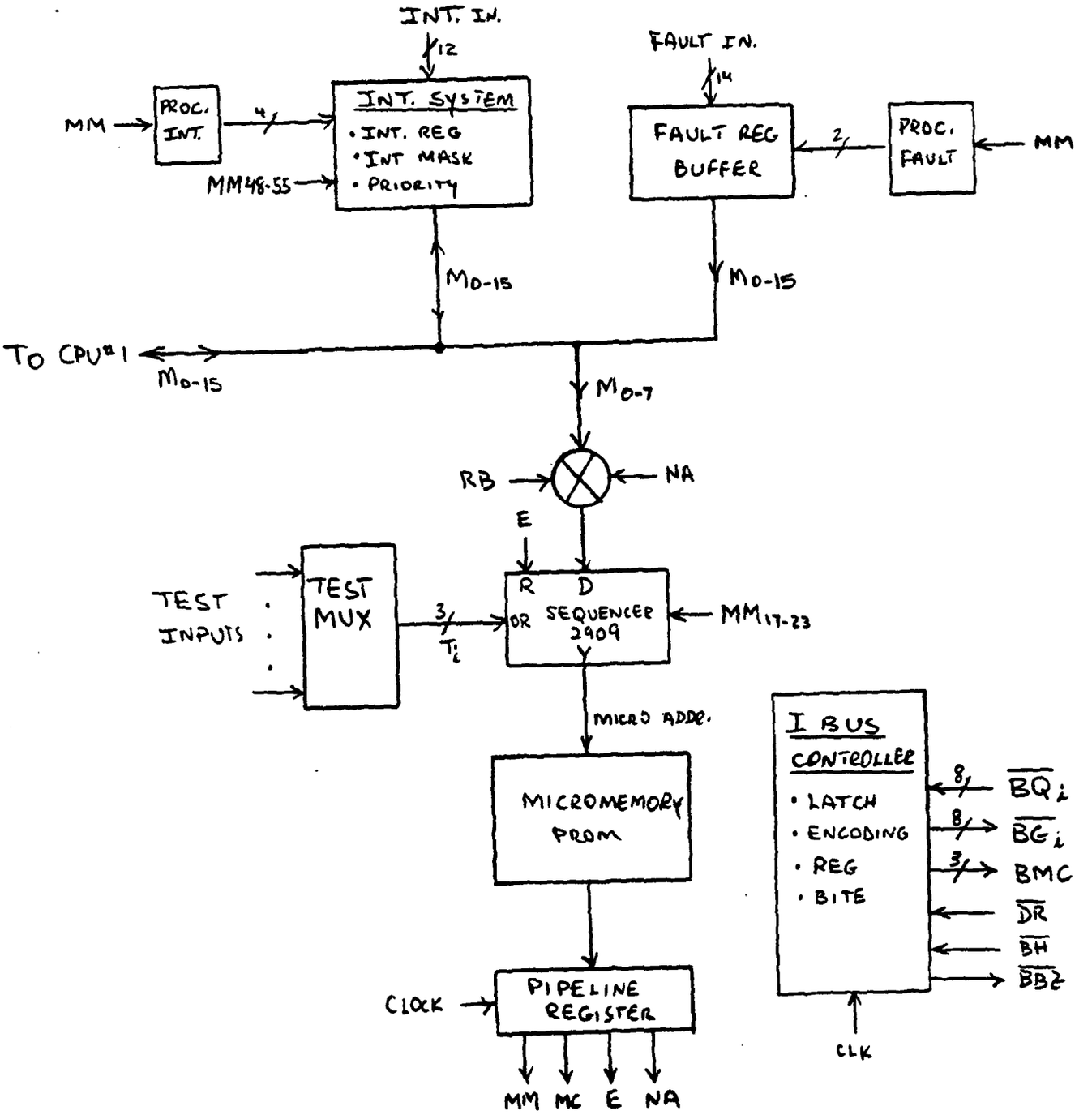


FIGURE 20. CPU #2 BLOCK DIAGRAM

Program Counter - This 16 bit register normally holds the address of the next instruction to be executed. It can also be gated onto the M bus with the two bytes of the word interchanged, as required for the XBR instruction.

Fast Multiply - This circuitry provides serial multiplication at a 10MHz bit rate. When the flip flop KS=1, the multiplier multiplies two 24-bit, 2's complement, fractional numbers; when KS=0, it multiplier two 16-bit integers. Integer multiplication may be performed with signed, 2's complement operands or with unsigned operands.

Memory/Input-Output Interface - All data transfers between the MC and external devices take place over the 16-bit, bi-directional Ibus. Data into the MC is gated from the Ibus to the M bus; data from the MC is gated from the S bus to the Ibus.

Addressing of external devices is defined by the 16-bit MAL bus. The address from the MC comes from the Memory Address Register (MAR), which is loaded under control of the microprogram. The Memory-I/O bit from the MC is used to distinguish between the memory and other input/output devices.

Timing and Control - This circuitry provides the clock signals required by the MC, and the initialization signals to assure orderly and predictable startup. Clock frequencies of 4MHz and 10MHz are generated for the various units of the microcomputer system. A clock for the ALU is also generated, but this clock has a variable period. Each microinstruction contains a field which selects a period of 200, 250, 300, or 350ns, the choice depends on the time required to execute that particular microinstruction. In addition, the ALU clock is delayed until any data transfer via the Ibus is completed, provided that transfer is to or from the ALU.

The logic reset signal from the power supply is used to initialize the flip flops and registers in the ALU. When the reset is active, the microinstruction is cleared to ZERO; when the reset is removed, the first microinstruction is obtained from location ZERO in the micro-memory.

Microprogram Control - The microprogram control section consists of a micro-memory, a pipeline register, and sequencing logic.

Microprogram Memory - The microprogram resides in a 1536-word by 64-bit memory. The memory is implemented using 24 512-word by 8-bit PROM's. Typical access time is 35ns.

Pipeline Register - This 64-bit register holds the microinstruction being executed. It is loaded from the output of the microprogram memory. In most cases, this allows calculation of the next microinstruction address and the next microprogram memory access to occur concurrently with execution of the current microinstruction.

Sequence Logic - The sequencing logic controls the addressing of the microprogram memory. Three AMD 2909 sequence chips provide the basic address control, including a 4-word subroutine return address stack. Logic is also provided external to the sequence chips so that the address can be provided by the M bus (for opcode decoding), the microinstruction, or the RB register (for secondary opcode decoding).

Conditional branching within the microprogram is implemented by OR-ing the test conditions into the three least significant address bits. Up to three tests can be specified in each microinstruction, permitting up ti 8-way branches.

Fault Register - The fault register is a 16-bit register used for storing machine error conditions. The "OR" of the fault register bits shall be used to generate the "machine error" interrupt. The fault register has the capability of being read or reset under program control via input/output instructions. A machine error condition causes the corresponding fault register bit to be set to a "one". Unused bits are set to a "zero". Fault register bit assignment are in accordance with Figure 21.

Interrupt System - A vectored priority interrupt system is provided in the CPU. The interrupt system accepts sixteen interrupts, performs priority encoding on pending requests and when interrupts are enabled by the software, saves and loads the Interrupt Mask, Status Word, and Instruction Counter using addresses read from the interrupt vector table in memory.

Upon acknowledging an interrupt, the microprogram performs the interrupt vectoring operation as illustrated in Figure 22. For each interrupt, there are two fixed memory locations in the "interrupt vector table". The first location, called the linkage pointer, contains the address of where to store the present state of the computer. The second location, called the service pointer, contains the address of the new state of the computer. The CPU resumes instruction execution, starting at the new address of the instruction counter. Further interrupts are disabled until another "Enable Interrupt" instruction is executed. Return from interrupt is accomplished by executing the "Load Status" instruction with the value of the linkage pointer as the address field.

FAULT REGISTER

MSB	0 1	2 3 4	5 6 7	8 9	10 11 12	13 14 15	LSB
	MEM PROTECT	PARITY ERRORS	I/O ERRORS	ILLEGAL	SPARE	BITE	

BIT	FUNCTION
0	Memory Protect Violation - CPU
1	Memory Protect Violation - BCIU or DMA
2	Memory Parity Error (Read)
3	PIO Channel Parity Error (Write)
4	DMA Channel Parity Error (Write)
5	Illegal I/O Command. Input command for output instruction or output command for input instruction
6	PIO Transmission Error
7	Spare
8	Illegal address; memory not present
9	Illegal operation code
10	Spare
11	Spare
12	Spare
13	Watchdog Timer Overflow
14	Bus Timeout
15	Spare

FIGURE 21. FAULT REGISTER BIT ASSIGNMENT

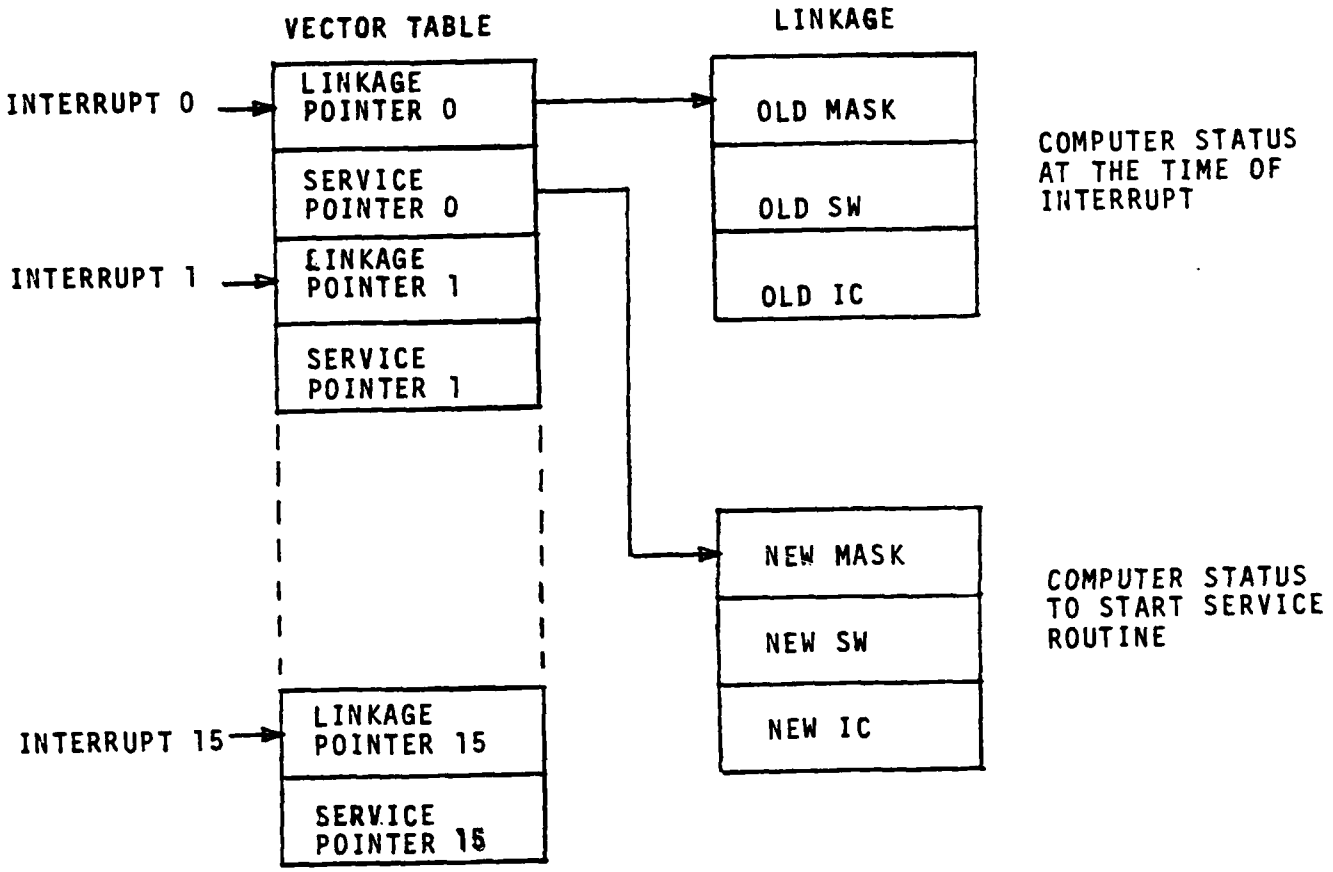


FIGURE 22. INTERRUPT SYSTEM

Prioritization of the sixteen interrupts is such that Interrupt 0 has the highest priority and Interrupt 15 has the lowest priority.

Internal interrupts occur for the following:

A. Illegal operation codes
B. Parity errors
C. Arithmetic overflow/underflow
D. Bus timeout
E. Memory Protect - from Memory
F. Interval Timers A and B - from I/O
G. Watchdog Timer - from I/O
H. Power Fail - from power supply.

Interrupt assignment and respective memory allocation for their pointers are in accordance with Table 14. The four bits representing the interrupt number are used as four bits of the 16-bit address of the linkage pointer.

All interrupts, except for interrupt 0 and 1, are not acknowledged by the CPU unless interrupts are enabled. Interrupts are only enabled under software control. Interrupts are disabled however under hardware or software control as follows:

A. Interrupts, except for interrupt 0, are disabled by the hardware following initialization via logic reset and remain disabled until software execution of an "Enable Interrupts" instruction.

B. Interrupts are disabled by the hardware following acknowledgement of an interrupt and remain disabled until software execution of an enable interrupts instruction. Interrupt 0 is disabled by the hardware following the acknowledgement of another interrupt until the microprogram completes the transfer of program control to the execution of the first instruction in that service routine.

C. Interrupts are disabled by the hardware until the instruction following an "Enable Interrupt" instruction has been executed.

D. All interrupts are disabled by the hardware as a function of an externally supplied interrupt inhibit signal used for testing purposes.

E. Interrupts, except for interrupt 0 and 1, are disabled under software control by execution of a "Disable Interrupt" instruction.

TABLE 14. INTERRUPT DEFINITIONS

Interrupt Number	Interrupt Mask Bit Number	Interrupt Linkage Pointer Address (HEX)	Interrupt Service Pointer Address (HEX)	Interrupt Name
0	0	20	21	Power Down (cannot be masked or disabled)
1	1	22	23	Machine Error (cannot be disabled)
2	2	24	25	Spare
3	3	26	27	Floating Point Overflow
4	4	28	29	Fixed Point Overflow
5	5	2A	2B	Spare
6	6	2C	2D	Floating Point Underflow
7	7	2E	2F	Timer A
8	8	30	31	Spare
9	9	32	33	Timer B
10	10	34	35	Spare
11	11	36	37	Spare
12	12	38	39	BCI Level 2
13	13	3A	3B	Spare
14	14	3C	3D	BCI Level 1
15	15	3E	3F	User Console Interface

NOTES:
- Interrupt Number 0 is highest priority. Priority decreases linearly with increasing interrupt number.
- All Spares are assumed to be external interrupts for hardware purposes.

Interrupt Mask - The interrupt system contains an Interrupt Mask Register which provides for individual interrupt masking under software control. The software has the capability to load and read the mask register via I/O instructions. Setting of a bit in the mask register enables the detection of the interrupt corresponding to that bit. Interrupt 0 is unmaskable.

Interrupt Requests - External interrupt requests must be presented to the microprocessor for no less than 0.4 microseconds and for no greater than 6.4 microseconds.

Interrupt Response Time - The time to respond to an interrupt which has been accepted for processing by the microprocessor does not exceed 80 microseconds.

Implementation - All of the above interrupt requirements are implemented with the AMD 2914 Interrupt Encoder and the microprogram. Some additional hardware is required to handle the disable and mask restrictions on Interrupt 0 and 1.

Execution Times - Typical execution times are given in Table 15.

Microinstruction Format - Each microinstruction consists of 64 bits, which are partitioned into 20 fields. Figure 23 shows the microinstruction word format and the functions performed by each field. In most cases, each field performs its function independently of the other fields, but in some cases there is interaction between fields. For example, the EMIT field usually serves as a constant within the microprogram, but, when the STACK field contains a FIVE, it is used to control the interrupt, fast multiplier, and the external file, etc.

Initialization - After turn-on, the microprogram begins by executing the microinstruction at location ZERO. The initial microinstructions clear the interrupts, disable all interrupts except 0 and 1, mask off all interrupts except 0, and set the Status Register to 2000. Also, the Memory Address Register (MAR) is set to ZERO, and the Program Counter (PC) is set to ONE in preparation for reading the first macroinstruction from location ZERO.

Instruction Access - When a macroinstruction is read from memory, the MAR contains the address of the instruction (or the address of the first word of a 2-word instruction), and the PC contains an address ONE greater than the MAR; at the end of the microinstruction, the PC is loaded into the MAR in preparation for reading the second word if a 2-word instruction is accessed.

	R	D	IM	IS	B	BX	I	ICR
LOAD								
SINGLE PRECISION	1.00	3.00 (3)	2.20	1.40 (4)	2.25	2.45	4.00 (3)	
DOUBLE PRECISION	1.40	4.20			3.25	3.65	5.20	
EXTENDED PRECISION		5.40						
UPPER BYTE		4.30 (3)					5.30 (3)	
LOWER BYTE		3.60 (3)					4.60 (3)	
STATUS		5.00 (3)						
MULTIPLE REGISTERS		2.00+1.0N						
STORE								
SINGLE PRECISION		3.00 (3)			2.25	2.45	4.00 (3)	
DOUBLE PRECISION		4.20			3.25 (3)	3.65	5.00 (3)	
EXTENDED PRECISION		5.20						
UPPER BYTE		5.05 (3)					6.05 (3)	
LOWER BYTE		4.60 (3)					5.60 (3)	
REGISTER THROUGH MASK		4.80 (3)						
CONSTANT		3.40					4.20 (3)	
MULTIPLE REGISTERS		2.60+1.0N						
MOVE MULTIPLE WORDS		2.3 +2.5N						
EXCHANGE BYTES	1.90							
EXCHANGE WORDS	1.60							
ALL								
SINGLE PRECISION	1.30	3.30 (3)	2.30	1.90	2.25	2.55		
DOUBLE PRECISION	2.00	4.60-4.80						
FLOATING	5.70 (8)	8.50 (8)			7.55 (8)	7.70 (8)		
EXTENDED FLOATING	7.75 (9)	11.15 (9)						
SUBTRACT								
SINGLE PRECISION	1.30	3.30 (3)	2.30	1.90	2.35	2.55		
DOUBLE PRECISION	2.00	4.60						
FLOATING	5.70 (8)	8.50 (8)			7.55 (8)	7.70 (8)		
EXTENDED FLOATING	7.75 (9)	11.15 (9)						

TABLE 15. EXECUTION TIMES

	K	D	1M	1S	B	BX	I	ICR
MULTIPLY								
SINGLE PRECISION-16 BIT	6.65	8.65	7.45	7.25 (5)				
SINGLE PRECISION-32 BIT	6.10	7.10	6.90		7.15	8.35		
DOUBLE PRECISION	25.95-26.25	28.75-29.05						
FLOATING	9.65	12.4			11.45	11.60		
EXTENDED FLOATING	44.40	48.0						
DIVIDE								
SINGLE PRECISION-16 BIT	8.40-8.70	10.2-10.5 (3)	8.40-8.70	8.8-9.1				
SINGLE PRECISION-32 BIT	8.2-8.5	10.0-10.3 (3)	9.0-9.3		9.25-9.55	9.45-9.75		
DOUBLE PRECISION	6.9+1.4N (6)	9.5+1.4N (6)						
FLOATING	31.85	34.45			33.65	33.85		
EXTENDED FLOATING	83.30	86.95						
BIT OPERATIONS								
SET/RESET BIT	2.50	5.30 (3)					6.30 (3)	
TEST BIT	2.75	4.75 (3,7)					5.75 (3,7)	
TEST & SET BIT		5.80						
SET/RESET VARIABLE BIT	3.10							
TEST VARIABLE BIT	3.35							
LOGICAL								
AND	1.20	3.20 (3)	2.20		2.25	2.45		
OR	1.20	3.20 (3)	2.20		2.25	2.45		
EXCLUSIVE OR	1.20	3.20 (3)	2.20					
NAND	1.40	3.40 (3)	2.40					

TABLE 15 (CONTINUED). EXECUTION TIMES

		R	D	IM	IS	B	BX	I	ICK
SHIFT									
SINGLE PRECISION		1.6+0.2N							
DOUBLE PRECISION		2.4+0.2N							
LOGICAL,CNT. IN REG.	-LEFT	2.65+.2N							
	-RIGHT	2.85+.2N							
ARITH.,CNT. IN REG.	-LEFT	2.65+.2N							
	-RIGHT	2.85+.2N							
CIRC.,CNT. IN REG.	-LEFT	2.65+.2N							
	-RIGHT	6.65-.2N							
DOUBLE,LOG.,CNT.IN REG.	-LEFT	3.25+.2N							
	-RIGHT	3.45+.2N							
DOUBLE,ARITH.,CNT. IN REG.	-LEFT	3.25+.2N							
	-RIGHT	3.45+.2N							
DOUBLE,CIRC.,CNT. IN REG.	-LEFT	3.25+.2N							
	-RIGHT	7.25-.2N							
COMPARE									
SINGLE PRECISION	(7)	1.40	3.40 (3)	2.40	2.0	2.35	2.55		
DOUBLE PRECISION	(7)	2.10	4.90						
FLOATING									
EXTENDED FLOATING									
BETWEEN LIMITS			3.9-5.5						
INCREMENT/DECREMENT									
INCREMENT MEMORY			4.1 (3)						
DECREMENT MEMORY			4.3 (3)						
JUMP & BRANCH									
BRANCH	(7)								
JUMP ON CONDITION	(7)								
SUBTRACT AND JUMP	(7)		2.0						
JUMP TO SUBROUTINE			2.1						
			2.65					2.9	1.38

TABLE 15 (CONTINUED). EXECUTION TIMES

	R	D	IM	IS	B	BX	I	ICR
STACK OPERATIONS								
STACK&JUMP TO SUBROUTINE		3.6						
UNSTACK & RETURN	2.4							
PUSH MULTIPLE REGISTERS	2.45+1.0N							
POP MULTIPLE REGISTERS	2.85+1.0N							
CONVERSION OPERATIONS								
FIX	2.70+.2N							
EXTENDED FIX	3.70+.2N							
FLOAT	2.05+.25N							
EXTENDED FLOAT	3.35+.25N							
ABSOLUTE VALUE	1.45							
DOUBLE ABSOLUTE VALUE	1.85							
FLOATING ABSOLUTE VALUE	3.85							
NEGATE	1.30							
DOUBLE NEGATE	2.00							
FLOATING NEGATE	3.60							
MISCELLANEOUS								
NO OPERATION	1.40							
BREAKPOINT	2.80/1.60							

NOTES:

(1) ALL TIMES ARE IN MICROSECONDS
(2) TIMES ASSUME 1.0 MICROSECOND MEMORY INCLUDING BUS
(3) ADD 0.2μS IF INDEXING IS REQUIRED
(4) ADD 0.2μS IF OPERAND IS NEGATIVE
(5) SUBTRACT 0.2μS IF OPERAND IS NEGATIVE
(6) N = NUMBER OF SHIFTS TO NORMALIZE THE DIVISOR;
ADD 0.2 μS FOR INDEXING
ADD 0.85μS IF N > 16
SUBTRACT 3.85μS IF 16 MSB'S OF DIVISOR ARE ZERO
(7) AVERAGE TIME; ASSUMES EQUAL PROBABILITIES
(8) ADD 0.2μS FOR EACH SHIFT TO ALIGN OPERANDS
ADD 0.25μS FOR EACH SHIFT TO NORMALIZE RESULT
(9) ADD 0.85μS FOR EACH SHIFT TO ALIGN OPERANDS
ADD 0.9μS FOR EACH SHIFT TO NORMALIZE RESULT

TABLE 15 (CONTINUED). EXECUTION TIMES

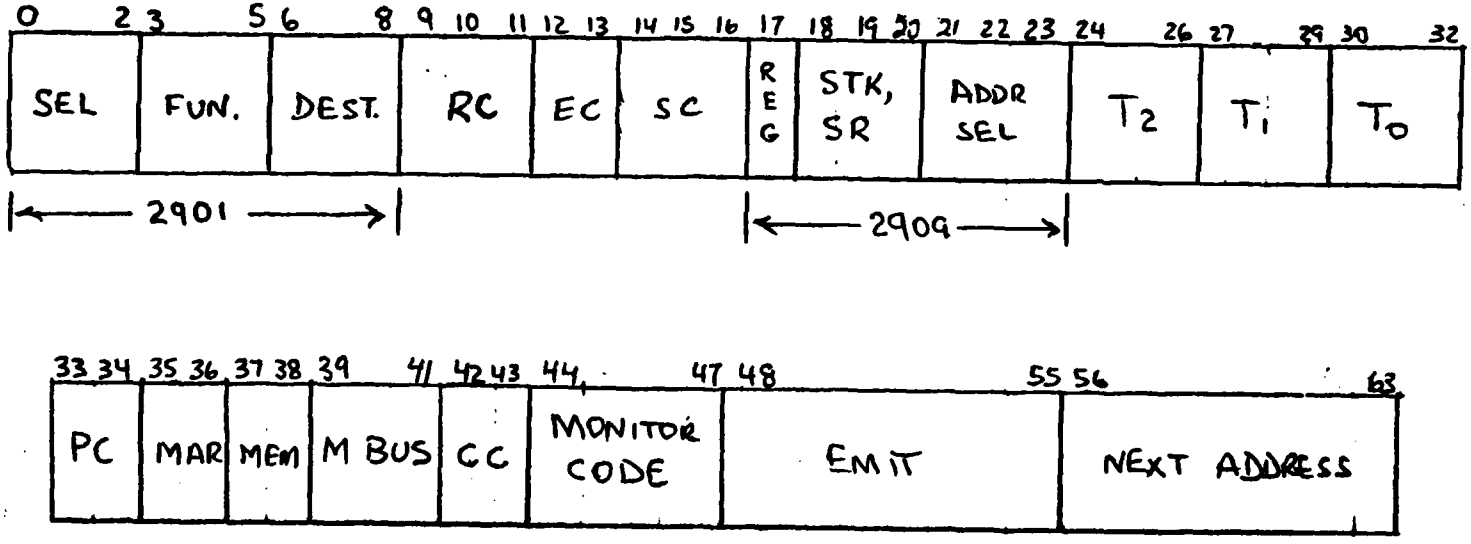


FIGURE 23. MICROINSTRUCTION WORD

10. MONITOR CODE

MM				FUNCTION	
44	45	46	47		
0	0	0	0	INSTRUCTION 1 READ	IN1
0	0	0	1	INSTRUCTION 2 READ	IN2
0	0	1	0	OPERAND 1 ACCESS	OP1
0	0	1	1	OPERAND 2 ACCESS	OP2
0	1	0	0	OPERAND 3 ACCESS	OP3
0	1	0	1	JUMP SOURCE	JS
0	1	1	0	JUMP DESTINATION	JD
0	1	1	1	DEFAULT - NO-OP	NOP (D)
1	0	0	0	INTERRUPT SOURCE	ISC
1	0	0	1	INTERRUPT DESTINATION	ID
1	0	1	0	BREAKPOINT	BP
1	0	1	1	SPARE	
1	1	0	0	SPARE	
1	1	0	1	SPARE	
1	1	1	0	INDIRECT EA	IA
1	1	1	1	SPARE	

FIGURE 23 (CONTINUED). MICROINSTRUCTION WORD

There is no instruction register per se; rather, each field of the accessed instruction is loaded into an appropriate MC register. Bits 6 and 7 (the base register) are loaded into the BR register, bits 8-11 into RA, bits 12-15 into RB, and bits 8-15 into the Step Counter. The contents of these fields are then used in an appropriate manner under microprogram control, depending on the particular type of instruction accessed.

The eight most significant bits of the instruction; i.e., the opcode, are used as part of the address of the next microinstruction to be executed. These addresses are determined by multiplying the 8-bit opcode by TWO and adding ONE. In the case of the Base Register format instructions where the opcode is only six bits, there are actually four addresses depending on the base register used.

If the STOP signal is active during an instruction access, then the next microinstruction address is not determined by the opcode, but comes from the Micro-Control Register in the 2909 chips. This, along with the STOP condition and the TO field, causes the microinstruction at address 3FB to be executed. When the STOP signal is turned off, the same macroinstruction will be accessed again. A similar procedure is used in the case of an interrupt, except that the next microinstruction address is 3F9.

Memory - The microprocessor is functionally capable of interfacing with any storage module meeting the requirements of Bus Interface.

Memory Capacity - The memory is capable of responding to up to 65536 addresses (excluding start-up).

Memory Word Length - Each word addressed contains sixteen data bits and one parity bit. The sum of the ones in the data field and the parity bit shall be an odd number. A read parity error is flagged as a fault condition with OMPE.

Memory Module Size - Each semiconductor memory module contains 1024 words or a multiple thereof. Each core module contains 16384 words or a multiple thereof.

Starting Addresses - Semiconductor memory may start at address 0, 16384, 32768, or 49152. Core memory may start at address 0, or 32768. Any mix of memory types is permitted so long as some module starts at address 0.

Memory Timing - No memory may have an access time greater than .450 microseconds nor a cycle time greater than one microsecond with respect to the falling edge of $\overline{IP}$ and the rising edge of $\overline{DR}$.

Start-Up ROM - There is a start-up version containing 2048 erasable semiconductor read only memory (ROM) for boot load capability.

Address Space - When the start-up ROM is enabled it responds to read requests at addresses 0 through 2047.

Start-Up State - Normal power start-up hardware initializes memory to the start-up state forcing instruction fetch from the start-up ROM. This state is maintained until a disable command is received from the microprocessor.

Memory Write Protect - Memory write protection is provided for the memory. This feature is software controllable in blocks of 1024 words (65536 words total), which shall be provided by means of a memory protect RAM (MPRAM). The MPRAM shall be loaded, read and enabled by I/O instructions. A logical one in the MPRAM indicates the protected state (for the associated block) and a logical zero indicates the unprotected state. All memory shall be protected after power-up until such time as the MPRAM is enabled, at which time the protection will be in accord with the MPRAM. Addresses 0 to 63 similarly prevent DMA write. During illegal write attempts, the memory will continue to handshake normally with the BUS MASTER. The memory will flag the error with $\overline{MPVC}$ or $\overline{MPVD}$ and not enable the addressed memory device. The contents of the addressed location remains unchanged. Separate flags are set for DMA/BCI errors and CPU errors.

Memory Commands - The following explicit input output commands pertain to memory:

	HEX Command	Mnemonic	Interpretation
a.	4003	MPEN	ENABLE MEMORY PROTECT RAM
b.	4004	ESUR	ENABLE START-UP RAM
c.	4005	DSUR	DISABLE START-UP RAM
d.	5000	LMP	LOAD MEMORY PROTECT RAM
e.	D000	RMP	READ MEMORY PROTECT RAM

Memory Configuration - 8192 words of random access semiconductor memory starting at address zero, 2048 words of erasable read only semiconductor memory, and 32768 words of core memory are implemented.

Memory Interface - The memory signals are illustrated in Figure 24.

MAL00-MAL15	These 16 lines from the bus master give the address to be accessed by memory.
I00-I15	These 16 lines are used to transfer data bi-directionally from the memory.
$\overline{IP}$	This line from a bus master indicates a valid address on the bus.
$\overline{ACK}$	This line indicates the address on the bus has been stored.
$\overline{DA}$	This line indicates valid data on the bus.
$\overline{DR}$	This line indicates the memory has stored data from the bus master.
$\overline{MINC}$	This line from the microcomputer inhibits memory operation.
$\overline{MINH}$	This line from the monitor inhibits memory operation.
M/IO	This line from the bus master indicates whether a memory or I/O operation is to be executed.
R/W	This line from the bus master indicates whether a read or write operation is to be executed.
$\overline{C10}$	This line from the microcomputer is the memory's 10MHz clock.
$\overline{ILA}$	Flag register bit from the memory which indicates the memory access of an unoccupied memory location.
$\overline{OMPE}$	Flag register bit from the memory which indicates a parity error occurred on a memory read.
$\overline{MPVC}$	Flag register bit from the memory which indicates the CPU tried to write into a protected memory block.

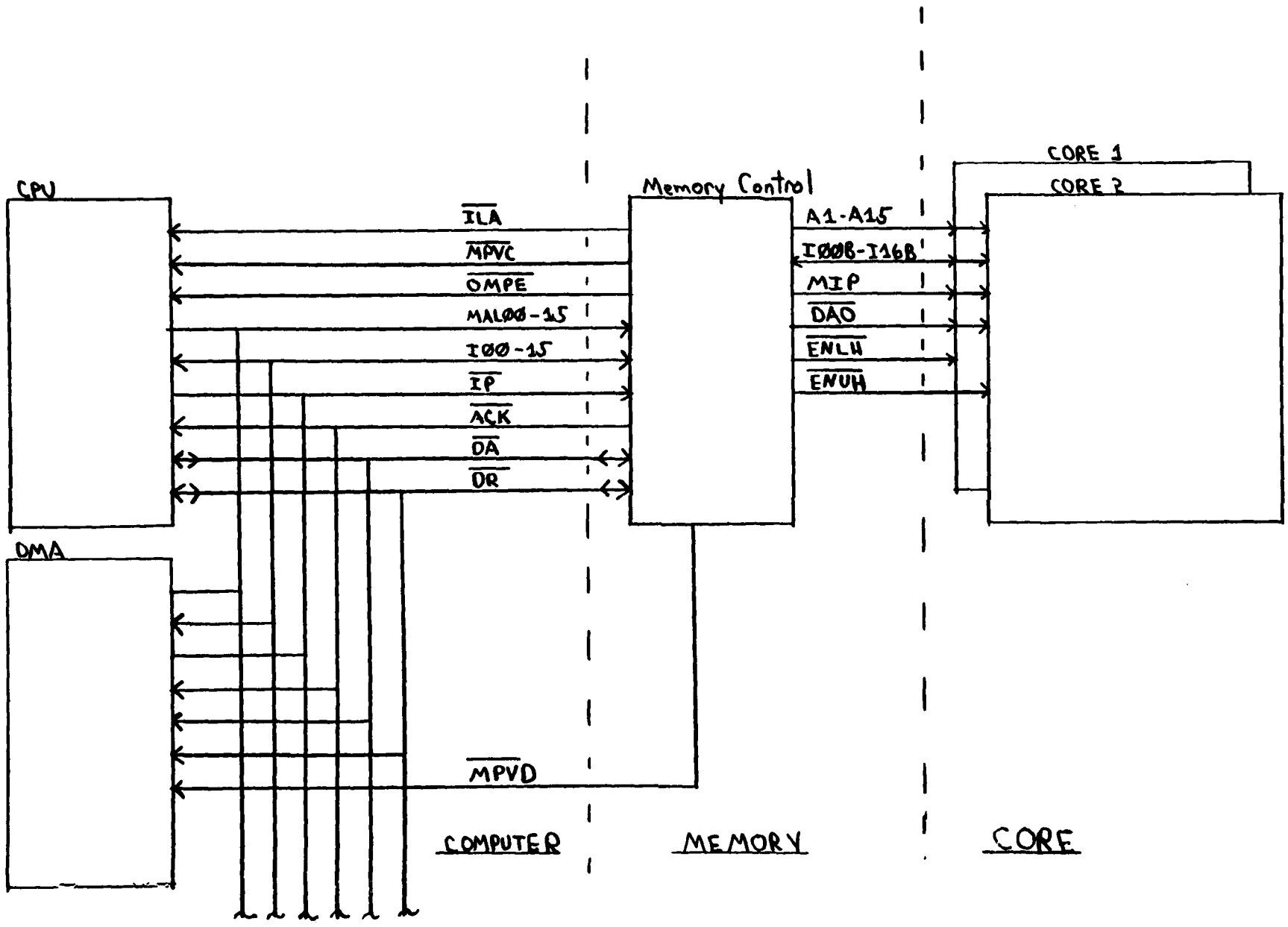


FIGURE 24. MEMORY SIGNAL INTERFACE

$\overline{\text{MPVD}}$	Flag register bit from the memory which indicates a bus master other than the CPU tried to write into a protected memory block.
$\overline{\text{LR}}$	This line from the monitor sets the memory into the prime state.
A1-A15	Internal memory address lines which interface to the microcomputer core.
$\overline{\text{ENLH}}$	This line from the memory enables core from 0 to 32767.
$\overline{\text{ENUH}}$	This line from the memory enables core from 32768 to 65535.
MIP	This line from the memory initiates a memory cycle in an enabled core.
MSBLSB	This line from the memory is high for a core read and low for a core write.
$\overline{\text{DAO}}$	This line from the memory enables the data buffers of a selected core.
I00B-I15B	These lines provide a bi-directional path for data between the core and memory.
I16B	This line is used to transfer parity bi-directionally between the core and memory.

Functional Description - The memory is physically partitioned into two sections. The first partition contains 32K of core with an option for an additional 32K of core and the required power supplies. The second partition contains semiconductor memory, timing and control. Figures 24 and 25 indicate the physical partitioning with a dotted line.

Core - The core section contains +15V power supply, +5V power supply, -12V power supply, signal and power cabling, mounting hardware for 2 SEMS16's, and 1 SEMS16 memory.

Core Configuration - All data, control, and address signals go to both the lower and upper memory modules with the exception of the most significant address which selects the memory modules.

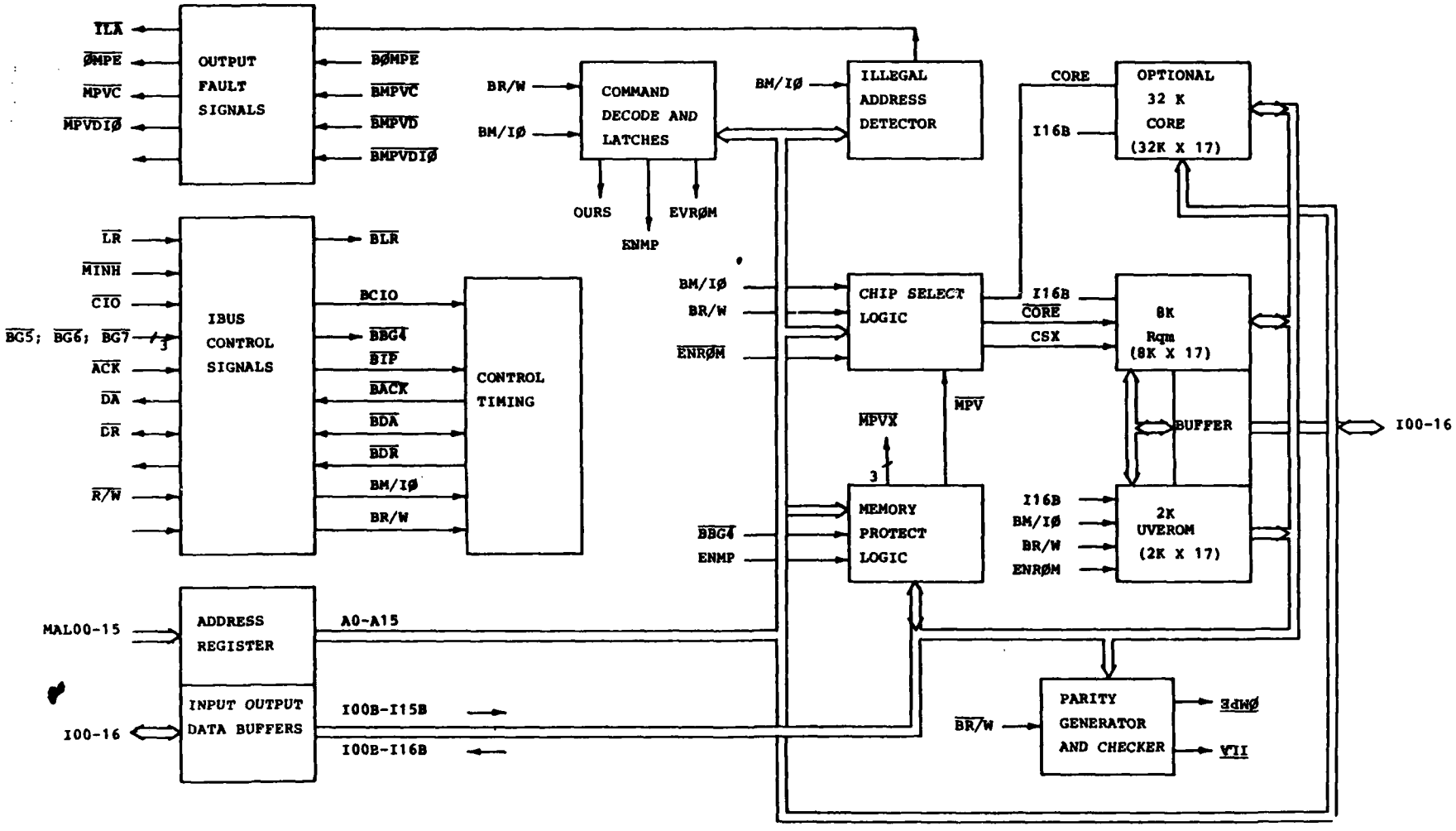
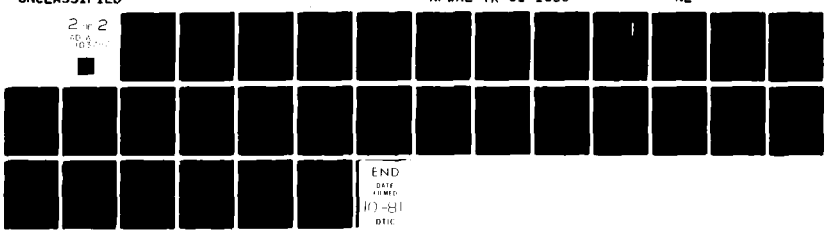


FIGURE 25. MEMORY FUNCTIONAL

AD-A103 782 SINGER CO WAYNE NJ F/G 9/2
A USER ORIENTED MICROCOMPUTER AND MONITOR SYSTEM.(U)
FEB 81 J W STARK F33615-77-C-1176
UNCLASSIFIED AFWAL-TR-81-1130 NL



Memory Control - The Memory Control section contains 2K EROM, 8K RAM, core interface, memory protection, parity generation, parity checking, illegal address detection, timing, computer bus interface, and I/O instruction decode section.

Start-Up ROM - The start-up ROM or shadow memory is 2048 words of ultra-violet erasable and electrically programmable read-only memory. This memory is enabled after system initialization. See Figure 26.

RAM - The 8K of RAM is divided into 2 4K blocks. These blocks are jumper selectable to any one of the 16 contiguous 4K blocks in the 64K memory space.

Core Interface - The core interface contains data buffering, parity generation, and control lines for two EMM SEM16's.

Memory Protect RAM - The memory protect is organized as 4 words of 16 bits for CPU write protection and 4 words of 16 bits for DMA write protection. All memory is protected after system initialization until the memory protect RAM is enabled. Memory is not protected from the system monitor writes.

Write Protect - The CPU write protect words reside at locations 00_{16} to 03_{16} and the DMA write protect words reside at locations 40_{16} to 43_{16} in the memory protect RAM. Each bit protects 1K of memory as illustrated in Figure 27. The MSB of word 00 in the memory protect RAM provides write protection from the CPU for the first 1K of main memory starting at location 0000_{16}.

Memory Violations - Attempts to write into a protected memory block will be flagged by fault register bits $\overline{\text{MPVC}}$ or $\overline{\text{MPVD}}$. The contents of the addressed location are not changed on a memory protect violation. The memory protect circuit checks the address early in a cycle. If the memory location is protected an enable to the addressed memory will not be generated. The normal Ibus interface control sequence will be followed during a memory protect violation.

Memory Parity - Parity is generated during a memory write and stored as bit I16 in main memory.

RAM Parity Detection - The RAM section contains a dedicated parity detector. During a RAM read I00BR-I16BR are gated through the detector. A parity flag will be issued if even parity is

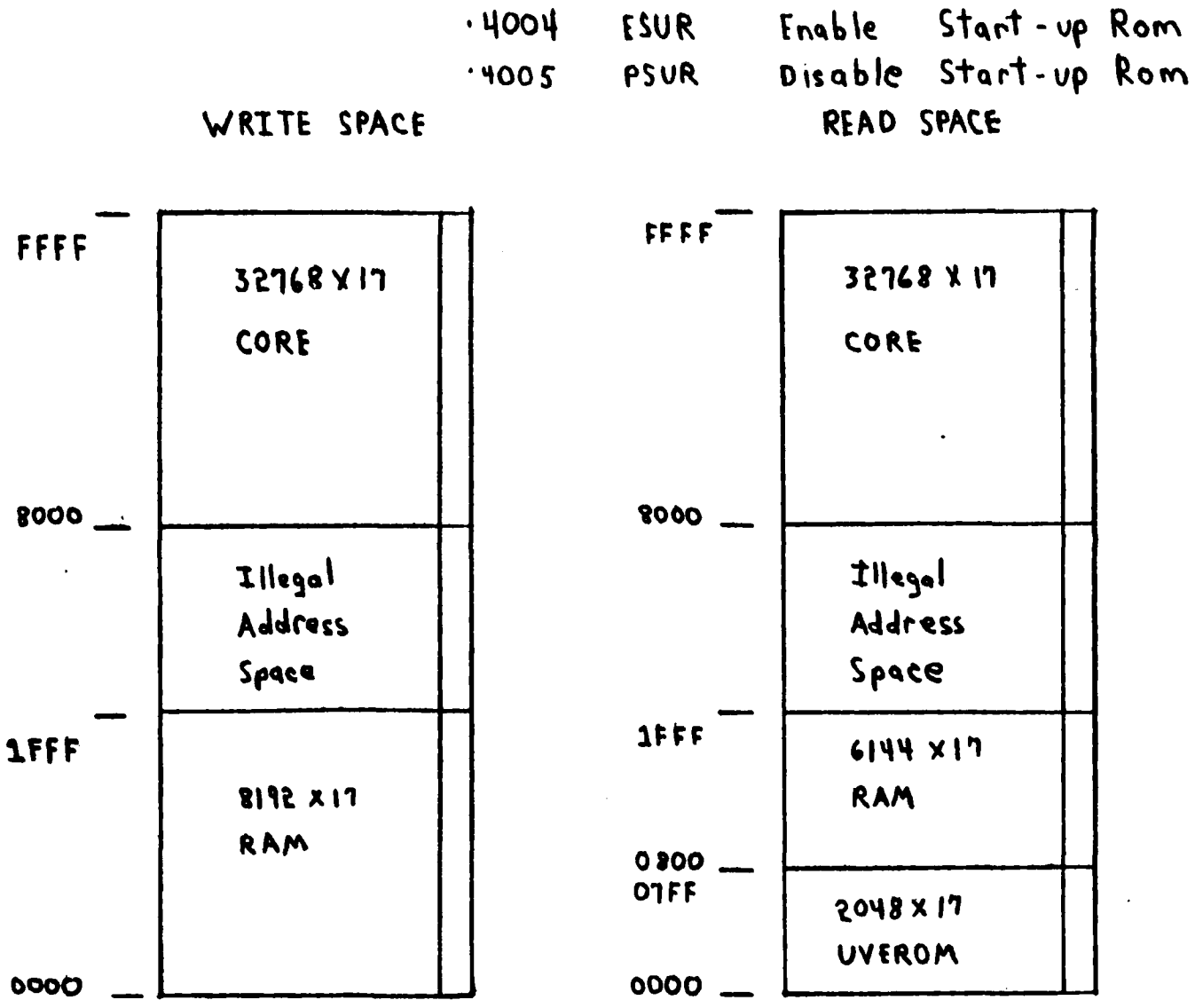


FIGURE 26. START-UP MEMORY ORGANIZATION

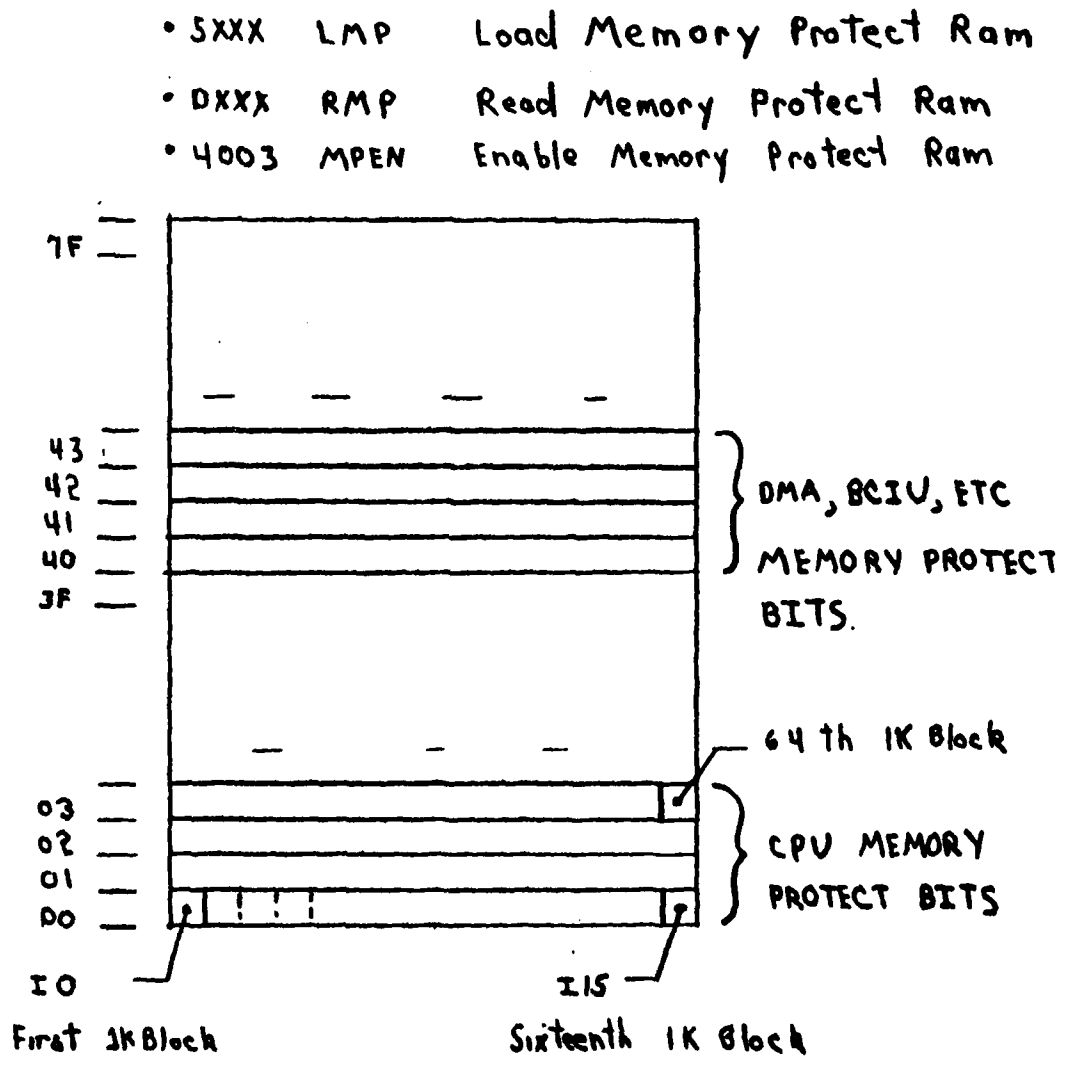


FIGURE 27. UOMMS MEMORY PROTECT RAM

detected. The parity flag is fault register bit $\overline{\text{OMPE}}$.

Core Parity Detection - The core section of main memory contains a parity generator and detector. During a memory read the parity generator output is compared with parity bit I16B. A parity flag will be issued if generated parity doesn't match stored parity bit I16B.

Illegal Address Detection - The illegal address detector will flag the attempt by a bus master to access on unoccupied memory location with fault register bit $\overline{\text{ILA}}$. The memory will continue to handshake normally with the bus master during an access to an unoccupied memory location.

Acceptable Illegal Addresses - The memory is divided into four 16K blocks which may in any combination be selected as illegal address regions. Further, one of these regions may be subdivided to specify a smaller region to the nearest 1K boundary from the upper 16K boundary.

Memory Timing - The main memory timing uses a "bucket brigade" logic implemented with J-K flip flops. The "brigade" contains appropriate wait states for the Ibus control sequence.

I-O Decode Logic - The I-O decode logic uses the address latched during an I-O cycle to enable the Ibus handshaking logic, gate write pulses to the memory protect RAM, enable memory protect output to the Ibus, set the memory protect flip flop, set the ROM flip flop, or reset the ROM flip flop.

Ibus Handshaking Logic - The handshaking logic uses the "bucket brigade" states to generate Ibus signals.

Input Output - The Input Output (I/O) consists of the following:

a. DMA Channel
b. Bus Controller Interface Unit
c. PIO Channel
d. Discrete I/O
e. Interval Timers
f. Watchdog Timer
g. Monitor Interface Buffer

DMA Channel - The DMA channel provides the capability to transfer 16 bit data words plus parity between the microcomputer and an external device. The DMA channel is under the control of the external device and memory transfers are accomplished on a cycle stealing basis. DMA write operations will not be permitted in protected memory areas. See Figure 28, DMA Channel.

The DMA is enabled, disabled, loaded, or read under control of I-O #1 microprogram control. The transfer of data between microcomputer memory and the external data is accomplished using a logic sequencer.

DMA Logic Sequencer - The DMA Logic Sequencer is a "bucket brigade" with appropriate wait states to meet Ibus specifications.

DMA Timing - The DMA timing is summarized below. The timing assumes no wait states.

DMA OPERATION	NS
READ	2300
WRITE	1700
ADDRESS ERROR	800
DATA ERROR	1500

Internal Interface Signals - In addition to the Internal Bus, the following signals are utilized.

- $\overline{C10}$ The ten megahertz clock is used for internal timing.
- $\overline{BRQ5}$ Bus Request Level Five is used to gain the internal bus.
- $\overline{BG5}$ Bus Grant Five enables the channel to control the bus.
- $\overline{DPE}$ Transmits the occurrence of a parity error on address or input data to the microprocessor.
- $\overline{LR}$ This signal is the system initialization.

DMA BITE - The DMA provides an internal wrap-around self-test capability which can be exercised under program control.

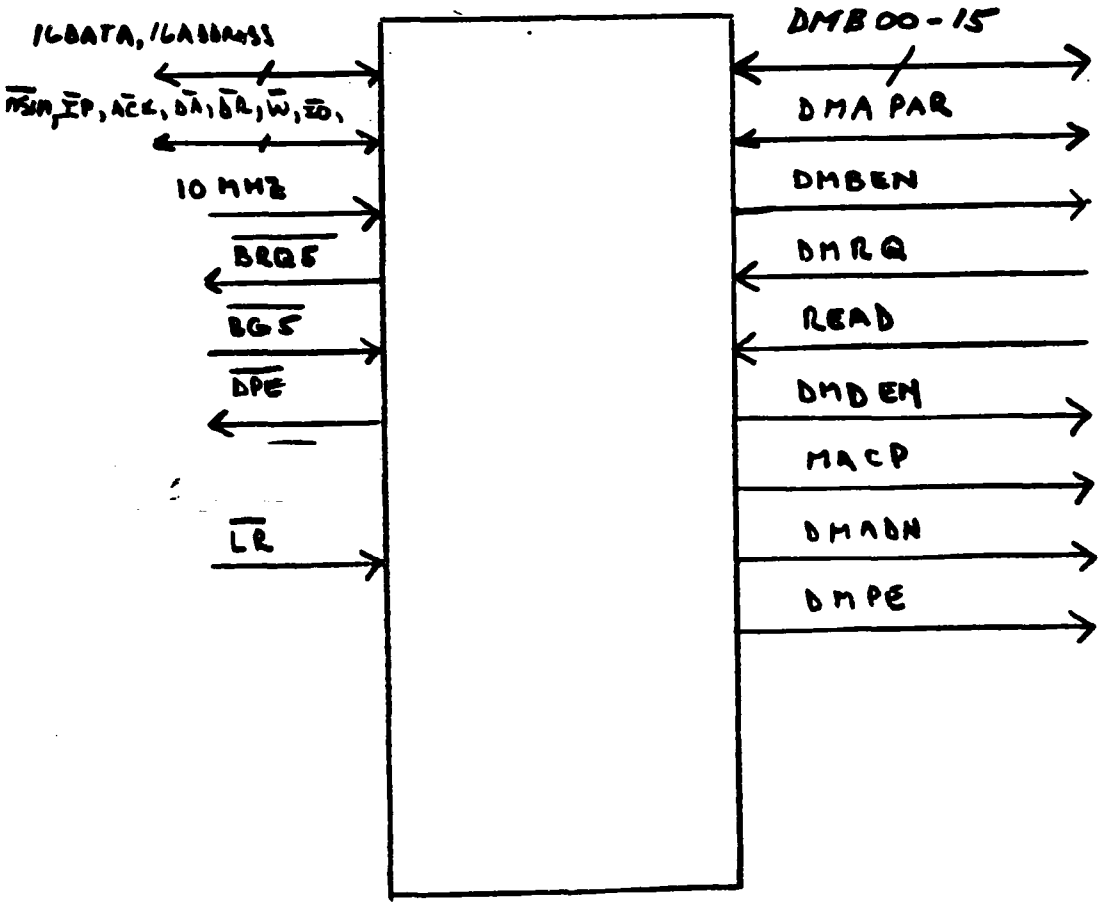


FIGURE 28. DMA CHANNEL

DMA Failure Conditions

If: Address (or Data) Parity Error on Read or Write

Then: Set DMADN
Set DMPE
Send DMA Parity Error to Fault Register in CPU

If: Attempt to Write in Protected Area

Then: Set DMADN
Send DMA Memory Protect Violation to Fault Register

Software - The DMA INO codes are:

INO Code	Mnemonic	Function
4006	DMAE	Enable DMA
4007	DMAD	Disable DMA
4002	DMSTL	DMA Self-Test - Load
C002	DMSTE	DMA Self-Test - Read

Monitor Simulation - To provide for monitor simulation, a latch is toggled when the monitor simulating line is pulsed and any of the DMA INO codes are detected. The latch is reset during power-up.

The Bus Controller Interface Unit (BCIU) - The BCIU functions in accordance with "Prime Item Development Specification for AN/AYK-15A Digital Processor", SA421205, 1 April 1979, Paragraph 3.2.1.4.

Figure 29 is a block diagram of the BCM portion of the BCIU and Figure 30 is a block diagram of the MTU portion of the BCIU. Figure 31 shows all the input-output signals for the BCIU and Figure 32 lists all the CPU instructions and their coding that relate to the BCIU.

The block diagram for the BCM, Figure 29, shows the various logical elements that control and process the various data flows between the MTU and CPU. Central to the organization of the BCM is the AMD 2901 microprocessor. It contains eight registers of sixteen bits each classified as "working registers" in the terminology of SA421205. The function of each of these registers is indicated in the section below entitled "BCM Microprogram". To

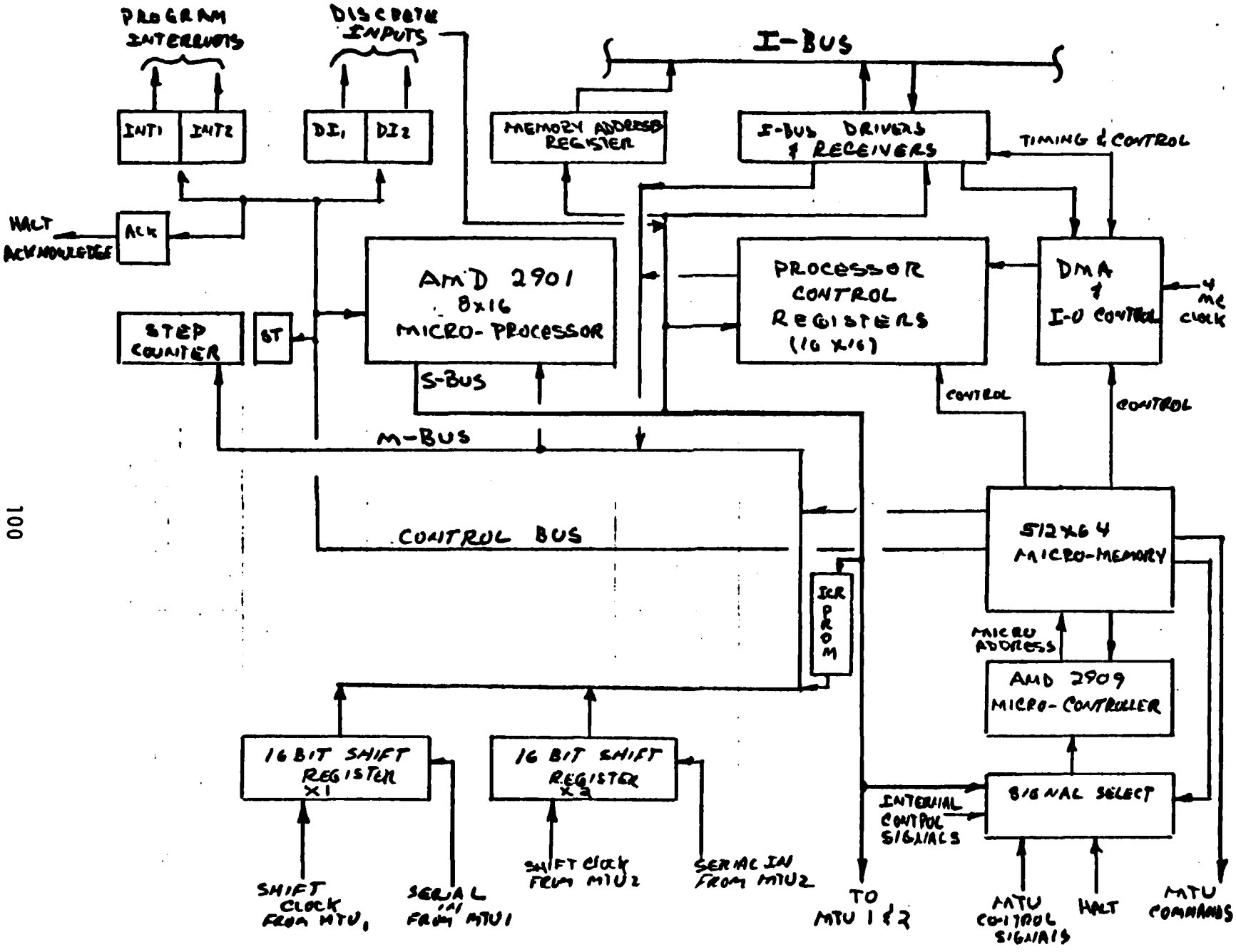


FIGURE 29. BLOCK DIAGRAM - BCM PORTION OF BCIU

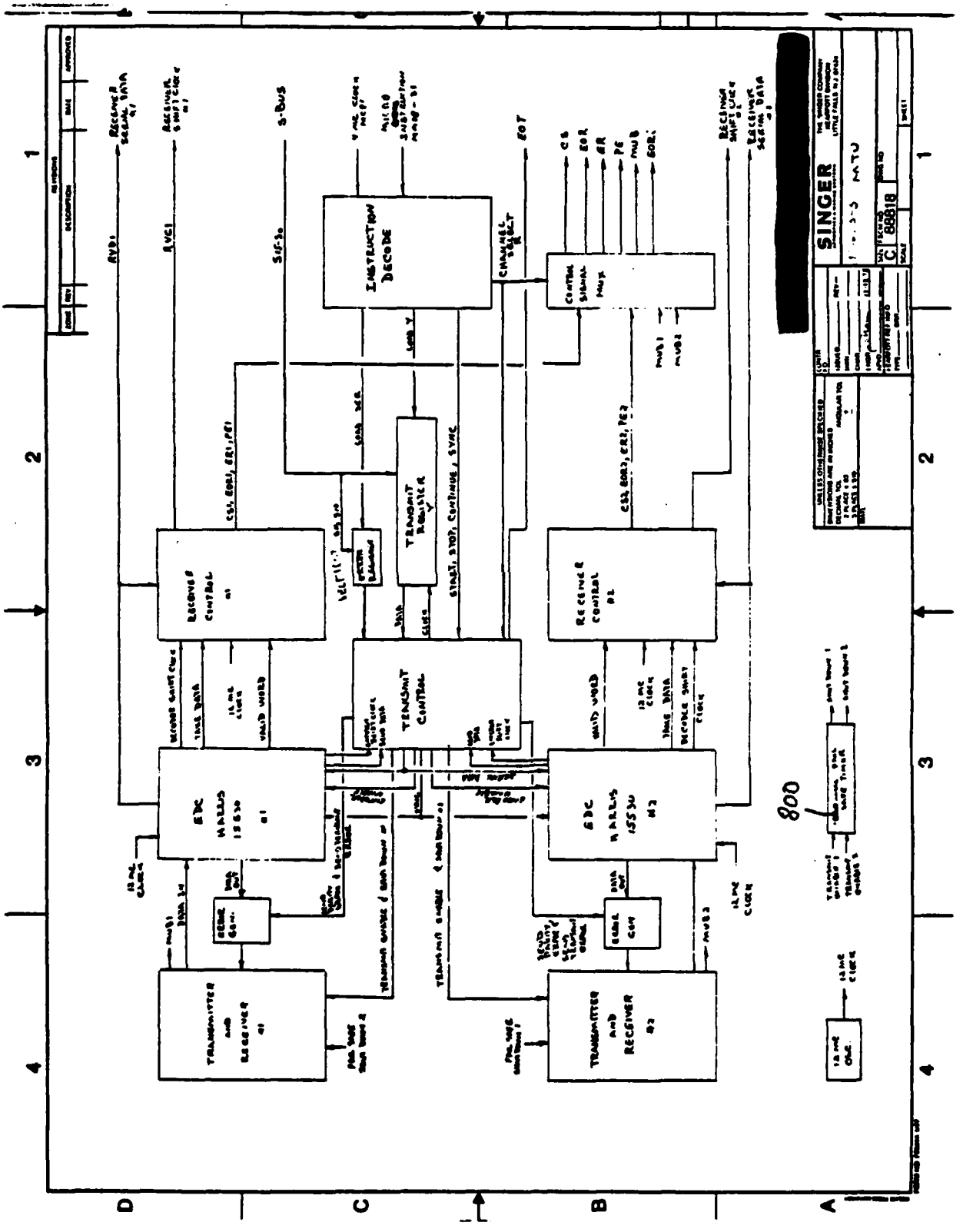


FIGURE 30. BLOCK DIAGRAM - MTU PORTION OF BCIU

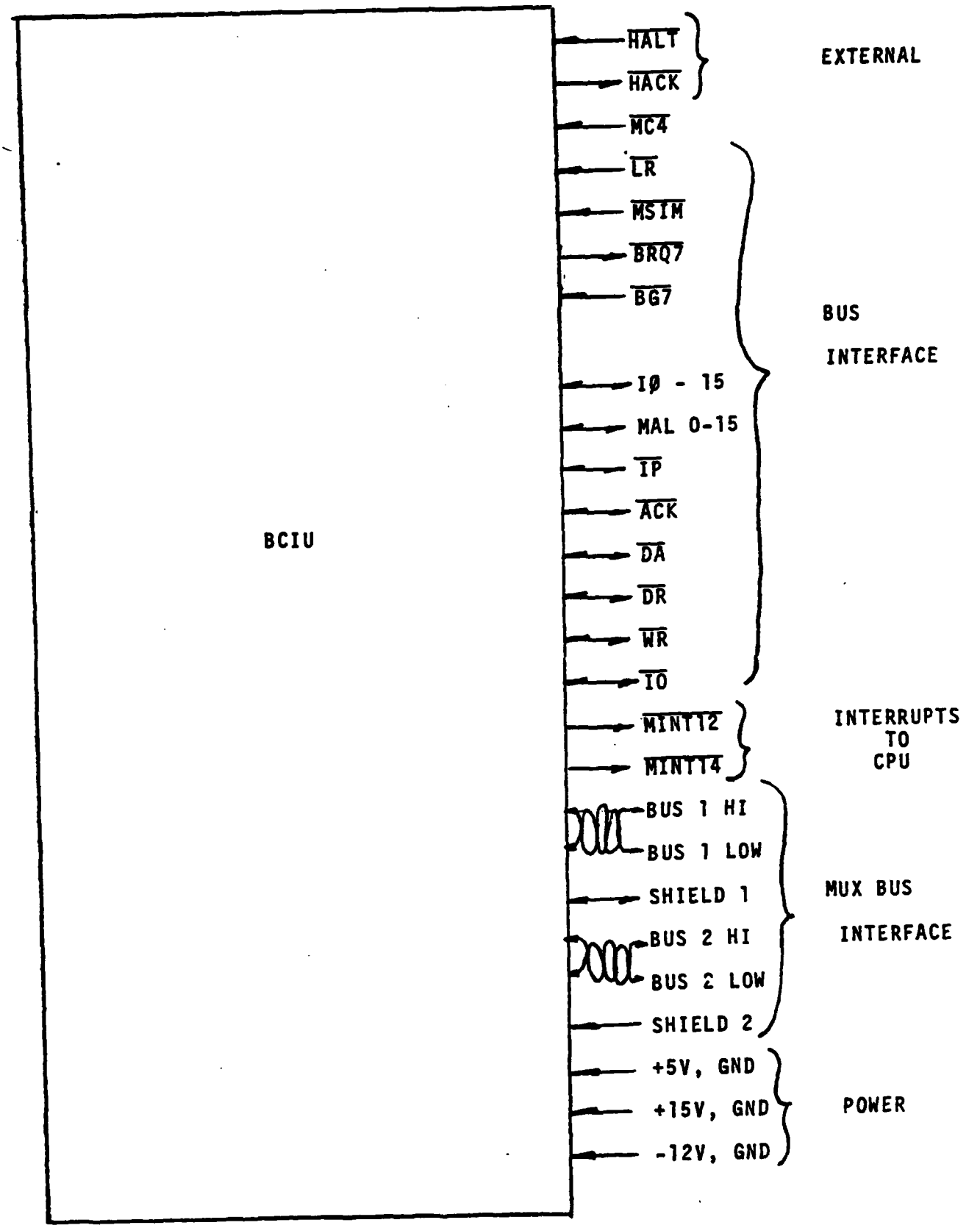


FIGURE 31. BCIU SIGNAL INTERFACE DIAGRAM

INSTRUCTION	FUNCTION	CODING			
		0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15
READ PROCESSOR CONTROL REGISTER	CONTENT OF THE SELECTED PROCESSOR CONTROL REGISTER TO CPU	1 0 1 1	0 0 0 0	0 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0	REGISTER SELECT
READ WORKING REGISTER	CONTENT OF THE SELECTED WORKING REGISTER TO CPU	1 0 1 1	0 0 0 0	0 0 0 1	0 REG. SELECT
READ DISCRETE INPUT ONE	DISCRETE INPUT ONE TO CPU IN L.S.B.	1 0 1 1	0 0 0 0	0 1 1 1	0 0 0 0
READ DISCRETE INPUT TWO	DISCRETE INPUT TWO TO CPU IN L.S.B	1 0 1 1	0 0 0 0	0 1 1 1	0 0 1 0
WRITE PROCESSOR CONTROL REGISTER	CPU TO SELECTED PROCESSOR CONTROL REGISTER	0 0 1 1	0 0 0 0	0 0 0 0 0 0 1 0 0 0 1 1	REGISTER SELECT
WRITE WORKING REGISTER	CPU TO SELECTED WORKING REGISTER	0 0 1 1	0 0 0 0	0 0 0 1	0 REG. SELECT
RESET DISCRETE INPUT ONE		0 0 1 1	0 0 0 0	0 1 1 1	0 0 0 1
RESET DISCRETE INPUT TWO		0 0 1 1	0 0 0 0	0 1 1 1	0 0 1 1

FIGURE 32. BCIU CPU INSTRUCTIONS

the right of the 2901 microprocessor, are the sixteen "Processor Control Registers", each sixteen bits wide. They hold the data that is received from and sent to the CPU-Memory under control of CPU-Instructions and the DMA facility of the BCM.

In the lower right corner are the microprogram memory and the AMD 2909 microcontroller used to generate the Micro Address for the microprogram memory. The microprogram memory contains 512 words of 64 bits each. Each word is one micro-instruction whose format and definitions are shown in Figure 33. This memory is a factory programmable read only memory.

Along the top is the circuitry for interfacing with the CPU and Memory. This includes the two program interrupts, the two discrete inputs, and the Ibus circuitry. The Ibus circuitry includes a sixteen bit memory address register and tri-state drivers, data tri-state drivers and receivers and the various timing signals for the Ibus such as DA, ACK, IP and DR.

At the center bottom are the two serial shift registers for converting the received serial data from the two channels of the MTU to parallel data.

In the center right is a 246x4 PROM used to convert the ISR indicators into the proper codes for the ICR in accordance with SA321200 and SA421205, paragraphs 3.2.1.4.2.4.13 and 3.2.1.4.2.6.12.

To the left of the 2901 microprocessor is an eight bit step counter used by the BCM for controlling the repeated execution of certain micro-instructions, generally those causing shifts of registers. Also located next to the step counter is a self-test flip-flop for use when the BCIU is performing its self-test in accordance with paragraph 3.2.1.6.2 of SA421205.

There are three main buses connecting the logical elements of the BCM. These are shown in Figure 29 and are called the S Bus, the M Bus, and the Control Bus. The S Bus is a sixteen bit data bus that carries the output of the 2901 microprocessor to the test of the BCM as well as the sixteen bit data word that is to be transmitted by the MTU to the MTU. The M Bus is a sixteen bit bus that carries various data into the 2901 microprocessor. The control bus is a 64 bit bus that distributes the BCM's micro-instruction to the various elements of the BCM.

Finally, it should be mentioned that the BCM logic is driven by a 4MC clock pulse supplied by the CPU.

Figure 30 shows the organization of the MTU section of the BCIU. It can be seen from the Figure that there are two channels of Transmitters and Receivers with a common control for transmission and separate control for reception. All words to be transmitted are loaded by the BCM over the S Bus into the Y register of the MTU. The Y register then converts this parallel word to a serial

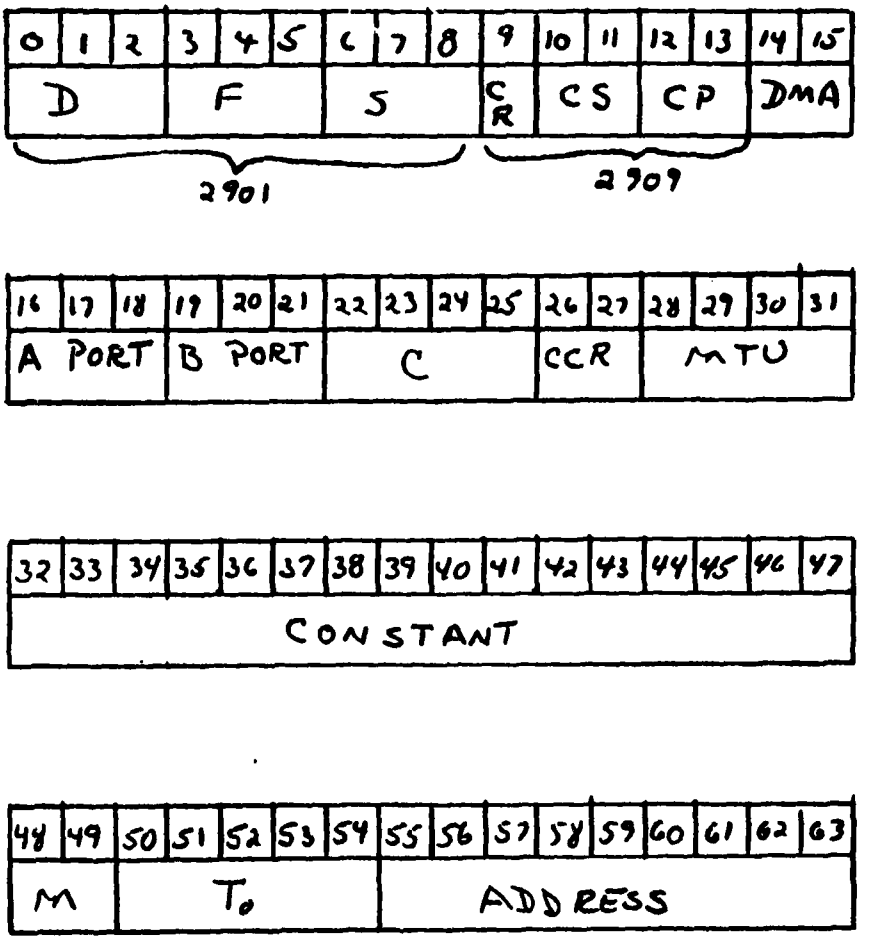


FIGURE 33. FORMAT OF THE MICROINSTRUCTION

word that drives the activated Harris EDC chip. Words received by the EDC chip are sent serially over the two serial lines to the BCM where they are accepted by the registers previously mentioned. Control of the MTU by the BCM is through one field of the BCM's micro-instruction: bits MM28, MM29, MM30 and MM31 seen in Figure 30 driving the instruction decode circuitry. Seven other signals from the MTU to the BCM provide current status of the MTU. These are the signals: (1) EOT: End Of Transmission, (2) CS: Command Sync Received, (3) EOR: End Of Word Receive, (4) ER: Validity Bit, (5) PE: Parity Error, (6) MUB: MUX Busy, and (7) EOR: End Of Word Receive Channel 1 or Channel 2.

The channel select signal, R, is a signal generated by the INSTRUCTION DECODE logic of the MTU and commands which channel is to be active.

For each of the two channels a small amount of circuitry is used to generate sync errors, parity errors and invalid Manchester under command of the BCM. This circuitry is shown in Figure 30 between the Transmitter-Receivers and the EDC chips.

As part of the MTU there is included an 800 usec fail safe timer which will shut down either of the two channels if either of them transmits for more than 800 usec.

Finally, a 12MC oscillator is included as part of the MTU to provide the EDC chips with the proper clock pulses.

Programmed I-O - Programmed I-O has been partitioned onto two logic cards. The first card contains the DMA Channel and Monitor Interface. The second card contains the PIO Channel, discrete I-O, Interval Timers, and Watchdog Timer. The names given to these cards are Monitor Interface and IO #1 respectively.

CPU Initiated Programmed I-O - All CPU initiated Programmed I-O has been realized by using a non-branching microprogram. The microprogram logic is contained on the IO #1 card.

Microprogram Fields - The microprogram fields are illustrated in Table 16.

- ACKR — If microcode bit 16 is active the microprogram state will not change until Ibus control line IP is active.
- DAR — If microcode bit 16 is active the microprogram state will not change until Ibus control line DR becomes active. Used to output DA on the Ibus.

TABLE 16. I-O MICROINSTRUCTION FIELDS

0	1	2	3	4	5	6	7	8	9	10	11	12 13 14	15	16	17 18 19 20 21	22
ACKR	DAR	DRR	DAT	CMDENR	DTENR	PBENR	XDAT	ENPD	ENPO	SITM	DMO	OUT	IBOUT	WAIT	STROBE	RCL/

- DRR — If microcode bit 16 is active the microprogram state will not change while Ibus control line DA is active.
- DAT — If microcode bit 16 is active the microprogram state will not change until Ibus control line DA becomes active.
- CMDENR — Used to strobe an address to external PIO device.
- DTENR — If microcode bit 16 is active and microcode bit 6 is not active, the microprogram state will not change until PIO bus signal XACK becomes active. This bit strobes data to the PIO channel on a write and terminates the PIO cycle on a read.
- PBENR — If microcode bit 16 is active and microcode bit 5 is not active, the microprogram state will not change until PIO bus signal XACK becomes active. This bit enables data from the PIO onto the PIO bus.
- XDAT — If microcode bit 16 is active the microprogram state will not change while PIO bus signal XACK is active.
- ENPD — Microcode bit used to enable data to the PIO output buffers.
- ENPO — Microcode bit used to enable address or data onto the PIO bus.
- SITM — Microcode bit used to select whether Timer A or Timer B is to be read.
- DMO — Microcode bit used to output DMA output register onto the Ibus.
- OUT — Three bit field used to select which output is to be read. Table 16 shows which I-O device is addressed.
- IBOUT — Microcode bit used to enable selected I-O device onto the Ibus.
- WAIT — Microcode bit used to stop the microprogram until there is a match between a microcode bit and an external control signal.

- STROBE — Five bit field used to generate a pulse to an appropriate device. Table 16 gives a short description for each field decode.

- RCL — Microcode bit used to set the I-O to the state where a new programmed I-O from the CPU will be accepted.

Wait Logic - The wait logic compares microcode bits with external control signals. When a match is not found between microcode bits and external control signals, the wait logic will inhibit clocks to the microcode counter. The inhibit logic uses control bit, GOB, to stop the divide by two microcode clocks.

The microcode clock is generated by counting two 100ns clocks. The inhibit logic stops the count after the first count and before the second clock. The microcode clock will occur a maximum of 100ns after the removal of GOB.

Microcode bit 16, WAIT or W, must be on for the wait logic to be enabled.

Programmed I-O Data Paths - The STROBE and Internal Control bits from the microcode output registers control the Inhibit logic, the state, and outputs of the functional elements shown in Figure 34. The microcode bits and STROBE decodes are shown in Table 17.

Ibus Interface Logic - The state machine of the interface is implemented by incrementing and stopping a 4-bit counter. The outputs are decoded and latched. During a bus transaction bus signal IP initiates the counter which increments to test state 4. If the decode is invalid the state machine waits for IP to terminate and resets to zero. If a code is valid at state 4 then ST4 starts the microprogram counter.

Interrupts - The I-O card contains 4 external interrupts. These interrupts are transmitted to the microcomputer over RS-422 compatible differential drivers.

Parallel I/O - The function of the parallel I/O (PIOU) is to transmit 16 bit parallel data words to, and receive data from, an external device under program control and to detect certain transmission errors when they occur. See Figure 35.

Interface - The PIOU interfaces with the internal bus to the microprocessor and to external devices. See Figure 36.

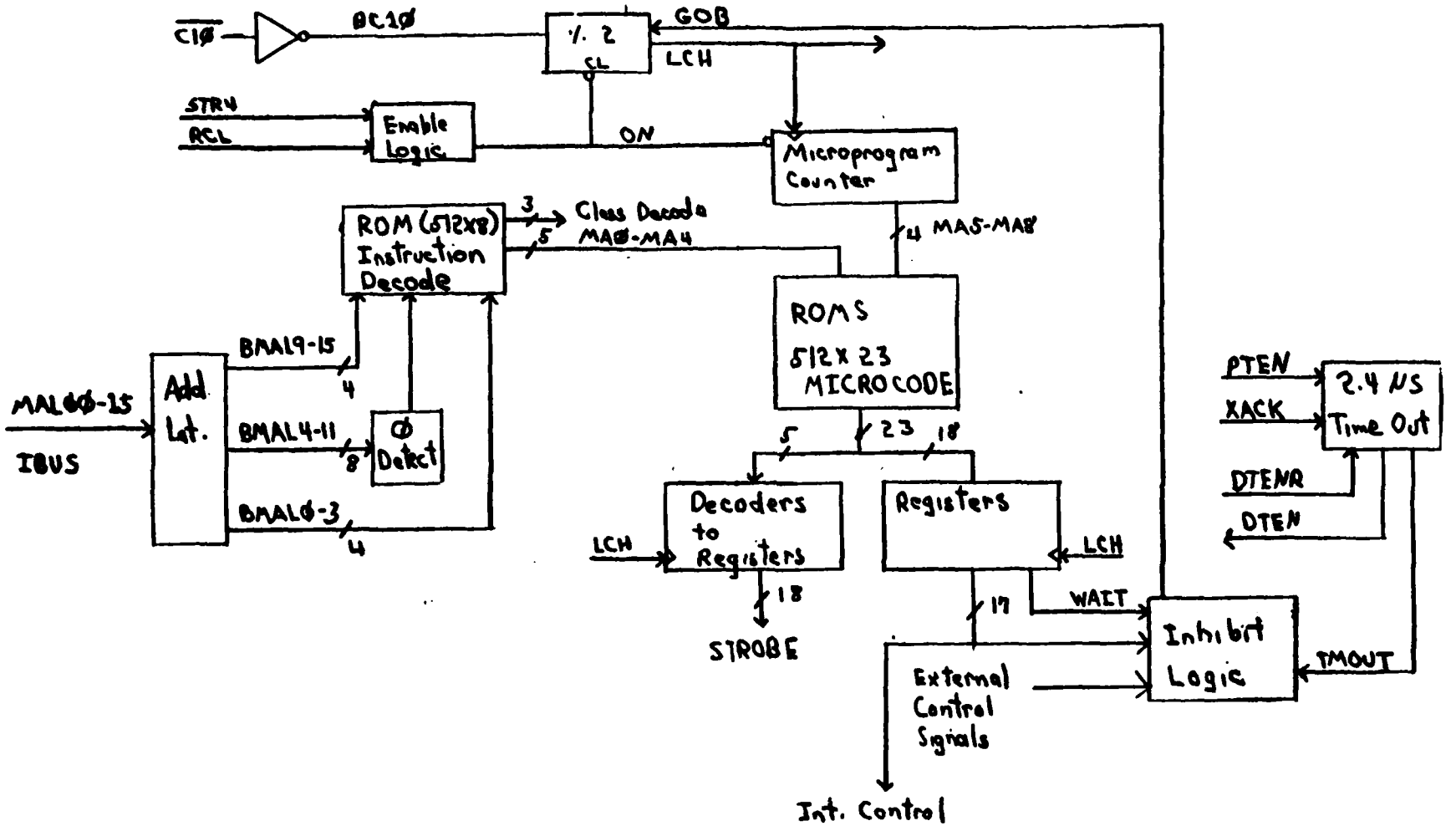


FIGURE 34. SEQUENCE CONTROL

MM

17	18	19	20	21	STROBE SIGNALS	MNEMONIC
0	0	0	0	0		
0	0	0	0	1		
0	0	0	1	0	TIMER A HALT	TAH/
0	0	0	1	1	TIMER A START	TAS/
0	0	1	0	0	LOAD TIMER A	OTA/
0	0	1	0	1	TIMER B HALT	TBH/
0	0	1	1	0	TIMER B START	TBS/
0	0	1	1	1	LOAD TIMER B	OTB/
0	1	0	0	0	RESET WATCHDOG TIMER	TGO/
0	1	0	0	1	WATCHDOG TIMER	WDTS/
0	1	0	1	0	LOAD PIO DATA REGISTER	LTPDR/
0	1	0	1	1	RESET POWER UP DISCRETE	DSI12RS/
0	1	1	0	0	LOAD DISCRETE OUTPUT REGISTER	LDSOR/
0	1	1	0	1	LOAD DMA OUTPUT REGISTER	LDMOR/
0	1	1	1	0	CLEAR PIO CHANNEL	DCLEN/
0	1	1	1	1	PIO TIMER ENABLE	DTEN/
1	0	0	0	0	DMA DISABLE	DMAD/
1	0	0	0	1	DMA ENABLE	DMAE/
1	0	0	1	0	HOLD TIMER A	HOLDA/
1	0	0	1	1	HOLD TIMER B	HOLDB/

MM

12	13	14	SELECT OUTPUT DEV.
0	0	0	
0	0	1	EITO/ (TIMER)
0	1	0	EPIB/ (PIO)
0	1	1	EDSIB (DISCRETE IN)
1	0	0	EDSOB (DISCRETE OUT)

TABLE 17. MICROPROGRAM FIELD DECODE

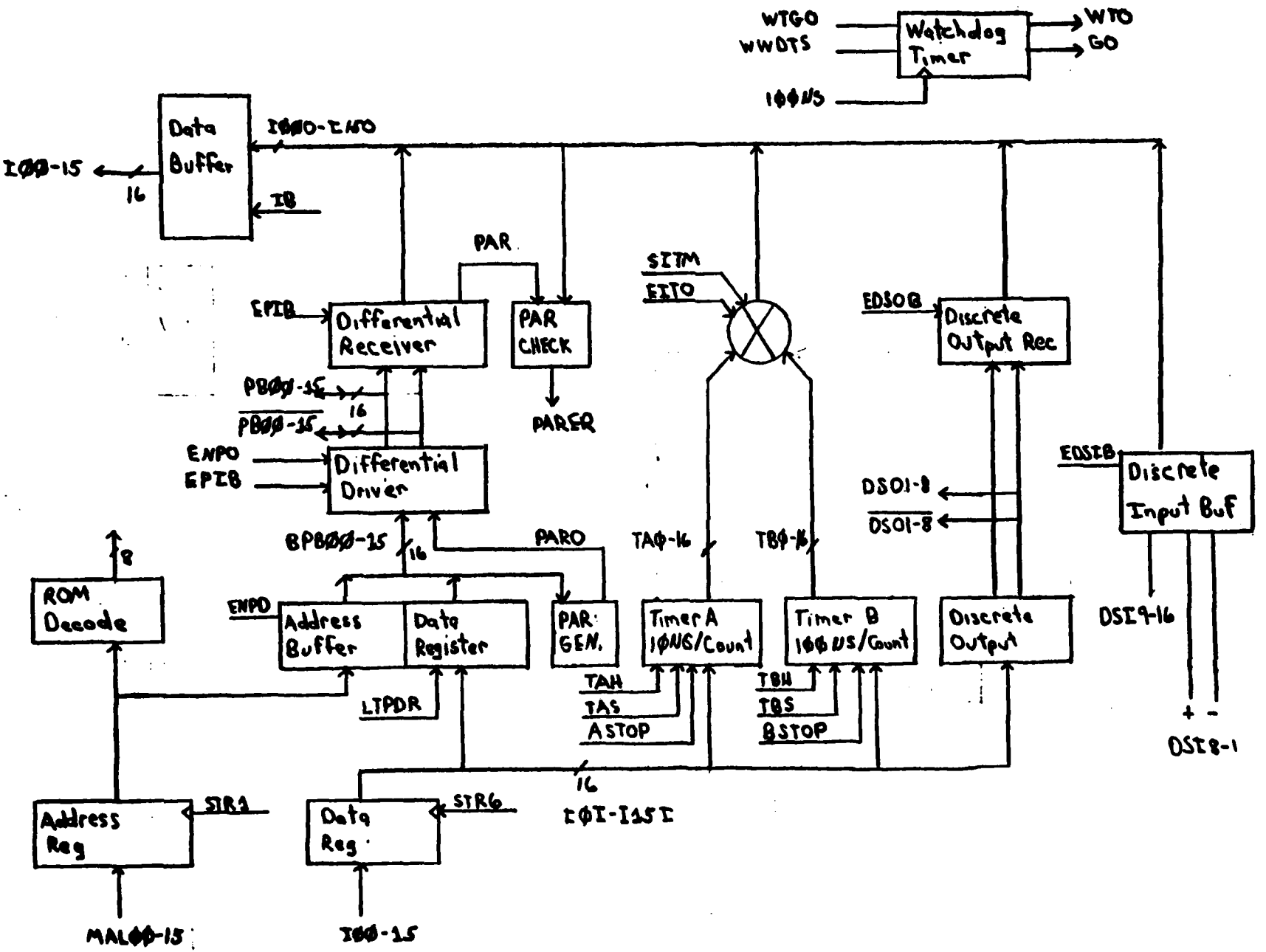


FIGURE 35. PROGRAM I-O ADDRESS AND DATA PATHS

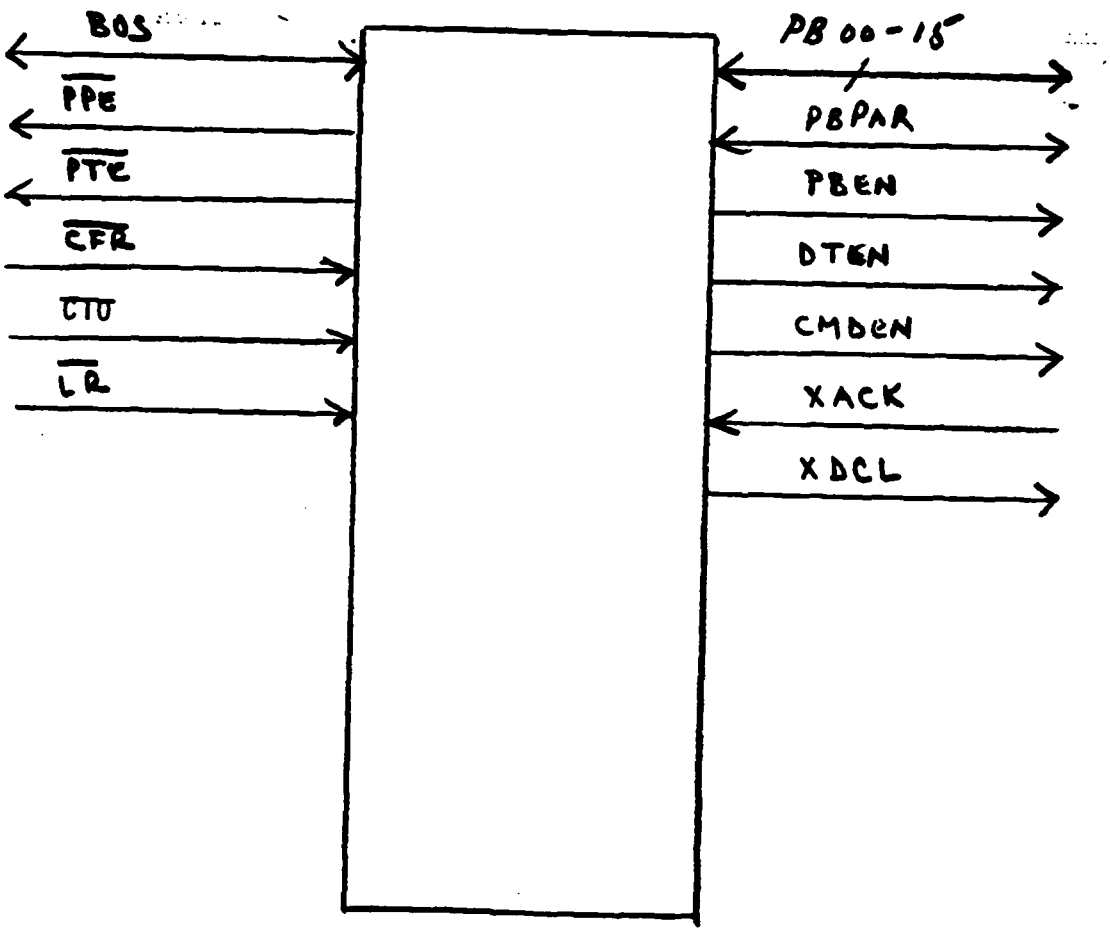


FIGURE 36. PARALLEL INPUT/OUTPUT

External Interface Signals - The PIOU external interface consists of 16 data lines, 1 parity line and 5 control lines.

Internal Interface Signals - In addition to the internal bus the following signals are provided:

$\overline{PPE}$ - An input odd parity error has occurred

$\overline{PTE}$ - A transmission error has occurred.

Output Transmission Error

If: External device does not respond with XACK,

Then: DTEN resets after 2.4 usec., and PIOU transmission error is sent to the Fault Register (in the CPU) and PIOU Busy remains set until reset by system clear or next correct transmission.

Input Transmission Error

If: External device does not respond with XACK,

Then: DTEN is not sent, and PIOU transmission error is sent to the Fault Register and PIOU Busy remains set until reset by system clear or next correct transmission.

Or If: A data parity error is detected,

Then: PIOU parity error shall be sent to the Fault Register.

$\overline{CFR}$ - Clear the fault register flags.

$\overline{C10}$ - This signal transmits the 10MHz clock.

$\overline{LR}$ - This signal is the system reset.

Software Interface - The PIOU INO codes are as follows:

INO Code	Mnemonic	PIOU FUNCTION
0000 + DEVICE ADDRESS	PO	PIOU OUTPUT
8000 + DEVICE ADDRESS	PI	PIOU INPUT
A00B	TPIO	READ PIOU OUTPUT REGISTER
200C	CPIO	CLEAR PIOU CHANNEL

PIOU External Interface Drivers/Receivers - All external interface drivers and receivers meet the requirements of EIA Standard RS422.

Monitor Simulation of PIOU Channel - To provide for monitor simulation, a toggle is set when the monitor simulating line is pulsed one and any PIOU code is detected. The latch is reset during power up.

Input Discretes - There are sixteen discrete input lines which can be sampled by input instruction. The discrete input line levels are set high and low at their source independent of software. Eight discretes are reserved for external inputs and the other eight for internal functions as shown. A high state line results in the corresponding bit of "one" when sampled, and the low state line shall result in the corresponding bit of "zero" when sampled. The external discrete signal line is +5V DC high differential line receivers which meet EIA Standard RS-422. The internal discrete lines are TTL compatible.

INO Codes and Data Formats - The input discretes have the following INO codes:

IND Codes	Mnemonic	Input Discrete Functions
A009	RPI	READ INPUT DISCRETES
200A	RNS	RESET NORMAL POWER UP DISCRETE

The input discretes format is as follows:

	MSB				INTERNAL							EXTERNAL				LSB
BUS BIT #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	DSI16	DSI15	DSI14	DSI13	DSI12	DSI11	PIOBUSY	DSI9	DSI8	DSI7	DSI6	DSI5	DSI4	DSI3	DSI2	DSI1

SHORT POWER TRANSIENT (UNIMPLEMENTED)

OVER TEMP (UNIMPLEMENTED)

NORMAL POWER UP

DISABLE INTERRUPTS (UNIMPLEMENTED)

UNASSIGNED

Output Discretes - The I/O has eight output discretes. An eight bit register is loaded by an output instruction and holds the discrete vector. A logical "one" is construed as the "on" state, and a logical "zero" as the "off" state, of the corresponding discrete. The discrete signals are initialized to "zero" at power up. The output signals are driven by +5V DC differential line drivers which meet EIA Standard RS-422. The ability to read block output discretes by a program input command is provided.

INO Codes and Formats - The INO codes for output discretes are as follows:

INO Code	Mnemonic	Output Discrete Function
2008	OD	LOAD OUTPUT DISCRETES
A008	RDOR	READ OUTPUT DISCRETES

The output discrete format is as follows:

	MSB															LSB
BUS BIT #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	NOT USED								DSO8	DSO7	DSO6	DSO5	DSO4	DSO3	DSO2	DSO1

Monitor Simulation - To provide for monitor simulation, a toggle is set when the monitor simulation line is pulsed and a load output discretes or read output discretes INO code is detected. When this latch is set, these two commands are not executed. The latch is reset during power up.

Other Discretes - In addition, a non-program controlled discrete for NO GO is provided.

Timer Input Output - The Timer Input Output provides two interval timers and one forty millisecond counter for system use.

Interval Timers - Two interval timers are provided, designated Timer A and Timer B. Timer A adds one count every 10 microseconds, and Timer B adds one count every 100 microseconds. Each timer is responsive to program output instructions to Load, Start or Stop, and to input instructions to Read. Each counter generates an interrupt request as its count goes from FFFF (hexadecimal) to 0000. If the counters are not loaded, an interrupt request is generated every 65536 counts. At power up, each timer is initialized to 0000 in running mode. The IDLE or TOINH line from the CPU, when set to logical one, stops both timers but does not reset them. When the IDLE and TOINH line is at logical zero, normal operation of the timers continues. The accuracy of the oscillator from which the timer pulse frequencies are obtained is ±0.01% or better.

Software Interface - The INO codes for the interval timers are as follows:

INO Code	Mnemonic	Timer Function
400A	OTA	LOAD TIMER A
C00A	ITA	READ TIMER A
4008	TAS	START TIMER A
4009	TAH	HALT TIMER A
400C	TBS	START TIMER B
400D	TBH	HALT TIMER B
400E	OTB	LEAD TIMER B
C00E	ITB	READ TIMER B

The interval timer outputs are formatted as follows:

	MSB														LSB
Bus Bit #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Go/No Go (Watchdog Timer) - A Go/No Go discrete output is provided which is driven by a counter which has a 40 millisecond timeout. When started, the output is high and remains high for 40ms at which time the output will go low. The counter is restarted by an output instruction. The counter is started as part of the power-up initialization. The idle input from the monitor when set to a logic one shall inhibit the timer but not reset it.

A short count bit test shall substitute a 10MHz clock for the normal 10KHz timer clock when the next programmed reset occurs, and so create a test timeout signal which is sent to the fault register in the CPU for BIT purposes. The Go/No Go discrete shall be unaffected by the test.

Software Interface - The IN0 command code is as follows:

IN0 Code	Mnemonic	Watchdog Timer Function
400B	GO	Trigger Go Indicator
400F	WOTSC	Start Count Watchdog Timer

Monitor Interface - The Monitor Interface provides EIA Standard RS-422 compatible drivers and receivers buffering the microcomputer bus and monitor control signals. In addition to the internal bus the following signals are provided:

$\overline{\text{BRQ1}}$	This is the monitor bus request signal.
$\overline{\text{BG1}}$	This is the bus grant to the monitor.
BMC0-2	These signals are an encoding of the current bus master identification.
$\overline{\text{BBZ}}$	This signal indicates bus status.
$\overline{\text{STOP}}$	This signal issued by the monitor halts the processor at the completion of the current instructions.
$\overline{\text{IDLE}}$	This signal issued by the microprocessor indicates that the MC has stopped.
MC0-3	These signals from the microcomputer identify what portion of instruction execution is in progress.
RA0-3	These encoded lines from the monitor identify which internal general register is to be transmitted to the monitor.

Monitor Interface Control - The monitor requires access to the Ibus under four conditions listed in the table below.

MONITOR STATE	INTERFACE CONTROL SIGNAL
MONITOR	DEFAULT
DMA	BG1, R/W
PANEL INSTRUCTION	MINH
MONITOR SIMULATION	TOINH

In the monitor mode the monitor receives all bi-directional signals. In the DMA mode the monitor responds as a bus master. In the Panel Instruction mode the monitor substitutes for memory. In the Monitor Simulation mode the monitors replace an I-O slave device.

The Monitor Interface asynchronously decodes the Interface Control Signals in less than 100ns. The decode selects the input or output driver of the required bi-directional signal.

I-O Timing - The following table lists the instruction times for the programmed I-O executed by I-O #1.

Command	Mnemonic	NS
400A	OTA	900
C00A	ITA	500
4008	TAS	500
4009	TAH	500
400C	TBS	500
400D	TBH	500
400E	OTB	900
C00E	ITB	500
400B	GO	500
400F	WOTSC	500
A009	RDI	500
200A	RNS	500
2008	OD	900
A008	RDOR	900
0XXX	PO	1500
8XXX	PI	1500
A00B	TPIO	900
200C	CPIO	500
4006	DMAE	500
4007	DMAD	500
4002	DMSTL	900
C002	DMSTR	900

Physical Description - The monitor and microcomputer are housed in standard laboratory "19 inch" consoles. The monitor console contains in separate drawers; the disk drives, the PDP 11/04 with its interface to the front end unit, the interface with the microcomputer, the balance of the front end logic, and power supplies mounted to the console frame. The console stands 63 inches high. The microcomputer console contains in separate drawers; two MIL Standard 1553B channels, the input/output and monitor port, the CPU and semiconductor memory, the core memory, and console mounted power supplies. (This console is 52" high.) The two consoles are interconnected by 24 foot long flat ribbon cables.

Each of the drawers except for the DEC supplied items was capable of supporting 4 Schottky type wire wrap boards. Each board in turn was capable of mounting over 150 dips. Interconnection between boards and drawers was predominantly provided by flat ribbon cables. Each board provides 12 26-pin connectors.

Approximate integrated circuit parts counts for major functions are given below:

Monitor

DEC Interface	41	
Micro Interface	81	
Micro DMA, Panel Instruction, Counters, Timers	129	
Bit Map	40	
Interrupt Logic, Trace Data Selection Logic	110	
Trace Buffer Controller, Value Comparison Logic	156	
3072x16 Trace Buffers	117	
	674	

Microcomputer

CPU	225	
Memory-8K RAM, 2K EVROM	136	
IO (except MUX)	183	
MUX Controller	128	
MUX R/T	21	+ 200 discrete
Monitor Port	61	
	754	

The core memory is a self-contained 32Kx18 SEMS 16 Memory.

THE BUILD AND TEST PHASE

The build and test phase required the building and testing of both software and hardware. A new assembler, linker, and loader accepting the Air Force assembly language, ALAP, producing appropriate output files, monitor resident software for servicing the user and driving the monitor front end, and various microcomputer test modules were required. The monitor and microcomputer logic needed wire wrapping, PROM conditioning and overall console and system test.

MIL-Standard-1750 and MIL-Standard-1553 microcode development were aided by the use of a Singer microassembler and simulator. Once micro-formats had been defined assembled microcode was tested against the macro (instruction level) definition using the micro simulator. Later the Air Force Mil-Standard-1750 design verification program and its associated data was used to drive the simulator for each of the MIL-Standard-1750 instructions. This greatly shortened the checkout of the final assembled CPU. The major CPU corrections required in the final hardware concerned the status register settings. Tests for proper status setting were not in early releases of the Air Force software and in some cases Singer had made a different interpretation of the expected setting.

The Air Force design verification program (AFDVP) was also used as the source input for the Singer support software to produce executable code for the microcomputer, in a file format consistent with monitor software requirements for loading and user symbol look-up. Since the UOMMS system was only configured with 40K of memory (8K semiconductor plus 32K core) the entire AFDVP was broken down to a simple executive and one test module per file. The decomposition worked well but expanded the AFDVP into well over one hundred files.

The major difficulties encountered with the overall system during integration were centered in the monitor mechanical configuration. The density of cabling and the attendent difficulty in trouble-shooting intensified the problem of debugging a state of art design. The cabling problem was recognized during the integration, and an alternative mechanical configuration was designed. Insufficient time was available, however, to make the transition.

Once the basic loading, panel instruction, and single step modes of the monitor were debugged the full checkout of the microcomputer proceeded rapidly and with a minimum of design problems in the microcomputer.

In the time remaining prior to a promised shipping date the balance of the features of the monitor were evaluated to the extent possible.

The various counters and timers were exercised. The stop conditions were checked. The bit map was loaded, read, and utilized with stopping and tracing, and data tracing was examined. Sufficient work was accomplished with tracing to insure proper operation of the buffer recording and downloading control logic and the proper capture of register data. Testing of traces based on operand value is incomplete.

Sufficient testing and analysis was completed to examine in general all of the proposed monitor system features and insure proper design and operation of the microcomputer prior to installation at WPAFB.

ACCEPTANCE AND EVALUATION PHASE

The UOMMS was installed at the Air Force Avionics Laboratory in June of 1980. Since that point in time it has been under evaluation by the Laboratory. Presently the user interface software is being rewritten by the Air Force to be compatible with the user interface developed under a later contract for other MIL-Standard-1750 computers.

CONCLUSIONS ON THE PRESENT IMPLEMENTATION

Medium scale integration, bit sliced, implementations of MIL-Standard-1750 and MIL-Standard-1553B are easily achievable but required a large number of circuits. Programmed logic arrays, more highly integrated chip sets, and VHSIC efforts will lessen the chip counts.

The idea of providing a monitor responsive to systems integrators and high order language programmers is viable. Responsiveness may be provided by a user interface capable of specifying run conditions and measurements with the symbols and data representations of his source listing. It is not self-evident as to whether the input specification should be in a command language or a prompting menu. The final prompting menus provided in this system vary from frames with little information content to overly filled frames. The frames also suffer from being full screen providing no insight into how one traversed to reach this frame (can you remember the last five frames?).

The monitor front end is hardware intensive containing over 600 integrated circuits. Non real-time functions, CPU and IO simulation, and performance measurement account for about one half the parts. Real-time data specification, recording, and downloading account for the balance.

The most useful information feature examined was the bit map. The bit map creates a Boolean function whose argument is absolute address and whose functional value is true if the programmer is interested in the particular absolute address and false otherwise. Its hardware implementation was a RAM memory with one bit in a word for full memory word in the microcomputer.

There is a very valuable tool for designating a set of addresses in advance of a program execution instance for breakpoints, locations to capture data, regions to count time. The function values are preset by the programmer. When the micro executes each address is presented to the bit map in real time when the bit map function is true the signal triggers the micro stop logic, or monitor data capture, or allows the counter of time.

The feature assumes even more power when software utilities are provided to allow symbolic specification. This feature was implemented for Singer symbols. Its power can be increased by allowing the utilities to specify regions as well as individual references.

This sort of mapping is also practical for data value identification though more complex for data items represented by multiple computer words.

Real time data capture or performance measurement requires a parallel interface to capture data or analyze activity.

If both address and data are of interest, even in this low performance CPU up to 2.5 million items per second are generated on the microcomputer bus.

The microcomputer bus was chosen as the listening point. This denied direct access to the internal registers of the CPU or any other module and eliminated real time access. All registers, not just the 16 register set, should be swept out. This should be generalized to any functional module.

An alternative implementation might be to look at the ALU output bus at times selected by a microcode field.

RECOMMENDATIONS

Sufficient work was accomplished on this contract to show the feasibility of developing embedded computers and their monitors so as to ease the task of Avionics Systems and software development and maintenance.

The necessary ingredients are: (1) support software that provides symbol table accessibility to monitor software, (2) monitor software that allows a user to examine and measure his system or program in language familiar to him (HOL level), and (3) a monitor/microcomputer interface with sufficient information content to allow discriminating, real time reporting.

The software ingredients would appear to be easily achieved. The monitor/microcomputer interface is an increasingly difficult challenge. This interface is not currently controlled by the MIL-Standard. Each computer vendor may vary his implementation of this interface depending upon his internal bus architecture. As even higher levels of miniaturization (VHSIC) come about pin limitations of the embedded computer (now on a chip) will limit still further the information available to monitor. This will come about even as applications become even more software intensive.

In the VHSIC technology, the practicality of incorporating the monitor function within the VHSIC chip itself should be examined as a possible solution. Specifying monitor function and reporting out might be through a very high speed on chip serial link. Capability required could be further explored using the present UOMMS.